

Geoinformatik 4

Geodateninfrastrukturen

Open Layers

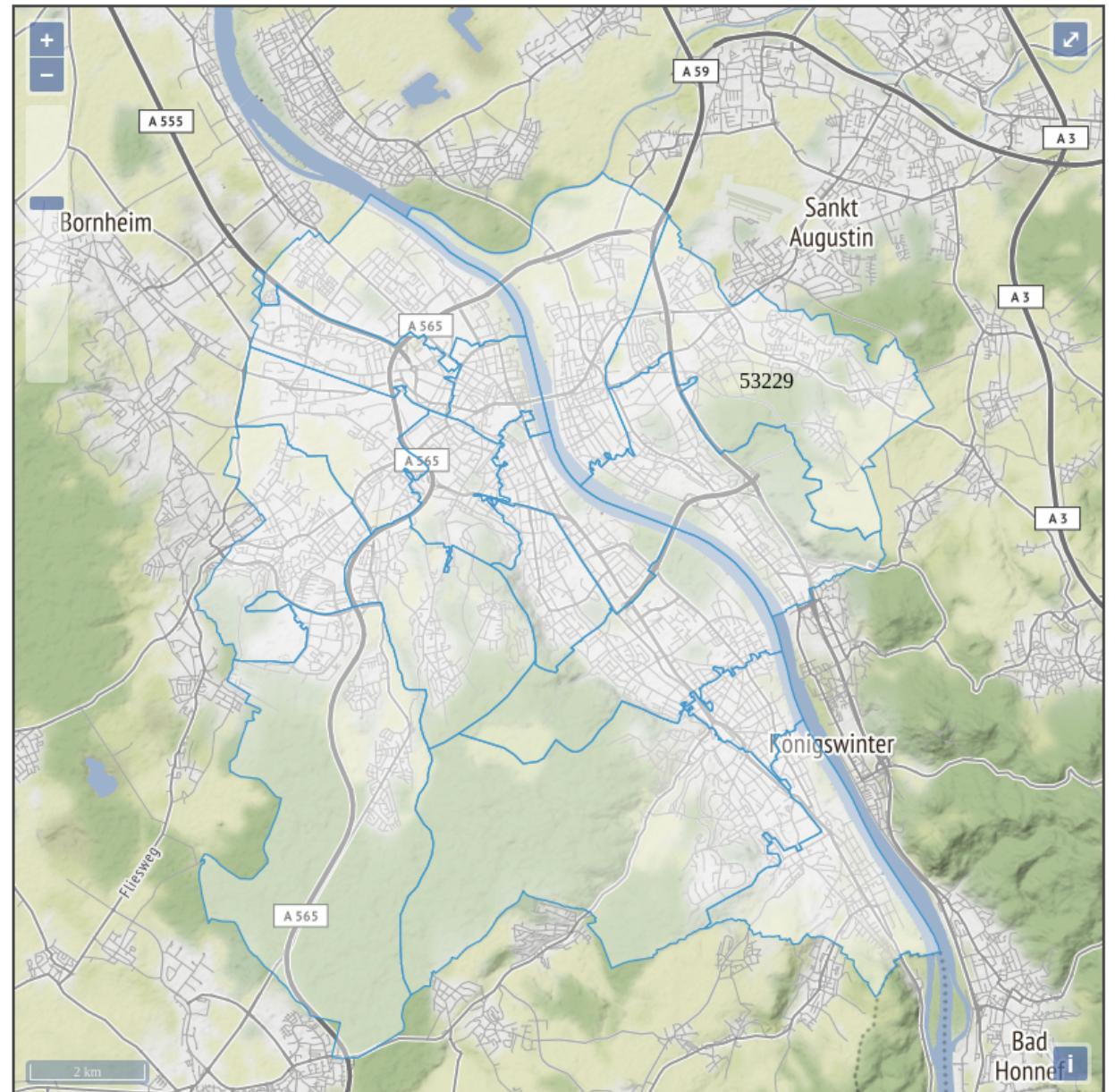
Prof. Dr.-Ing. Youness Dehbi
Computational Methods Lab
HafenCity University Hamburg

Ziel der Übung

Heute:

Interaktive Karte mit

- ▷ *HTML* (Hypertext Markup Language)
- ▷ *JS* (JavaScript)
→ **OpenLayers**



Verwendung von **JavaScript** im HTML-Dokument ermöglicht
Einbettung von **Karteninhalten**:

- ▷ **OpenLayers**
- ▷ ArcGIS API
- ▷ GoogleMaps API
- ▷ Geodaten über Webservices (WMS)
- ▷ Geodaten über Dateien (KML)

Verwendung von **JavaScript** im HTML-Dokument ermöglicht
Einbettung von **Karteninhalten**:

OpenLayers

- ▷ frei verfügbare Kartenbibliothek zum Einbinden von Geodaten:
 - Kartendienste (z.B. OpenStreetMap, Bing)
 - OpenGIS Web Services (OWS) (z.B. WMS, WFS)
 - Vektordaten (z.B. KML, GeoJSON)
- ▷ weitere Funktionalitäten wie Navigationselemente, Ereignisverarbeitung, Projektionen etc.

Aufbau

Einbinden von OpenLayers

```

<html>
  <head>
    <title>Beispielkarte Köln</title>
    <link rel="stylesheet" href="ol.css" type="text/css">
    <script language="javascript" Type="text/javascript" src="ol.js"></script>
    <style>
      .header {
        width: 800px;
        height: 50px;
        font-size: 20px;
      }
      .map {
        margin-top: 10px;
        width: 800px;
        height: 800px;
        border: 2px solid rgb(69,69,69);
      }
    </style>
  </head>
  <body>
    <div id="header" class="header"> Dies ist eine Überschrift
     </div>
    <div id="map" class="map"></div>
  </body>
  <script>
    // Layer mit Datenquelle
    var osmLayer = new ol.layer.Tile({
      source: new ol.source.OSM()
    });

    // Kartenmitte: Kovertierung in Bezugssystem von OSM (3857: Pseudo-Mercator)
    var centerPoint = ol.proj.transform([6.958281, 50.941278], 'EPSG:4326', 'EPSG:3857');

    // Festlegen der Kartenansicht
    var view = new ol.View({
      center: centerPoint,
      zoom: 12,
      projection: 'EPSG:3857',
      maxZoom: 16,
      minZoom: 4,
    });

    // Kartenerstellung und Verknüpfung zu HTML
    var map = new ol.Map({
      view: view,
      layers: [osmLayer],
      target: 'map'
    });
  </script>
</html>

```

HTML Stylesheet

HTML Inhalt

JavaScript

Einbinden von OpenLayers

```

<html>
  <head>
    <title>Beispielkarte Köln</title>
    <link rel="stylesheet" href="ol.css" type="text/css">
    <script language="javascript" Type="text/javascript" src="ol.js"></script>
    <style>
      .header {
        width: 800px;
        height: 50px;
        font-size: 20px;
      }
      .map {
        margin-top: 10px;
        width: 800px;
        height: 800px;
        border: 2px solid rgb(69,69,69);
      }
    </style>
  </head>
  <body>
    <div id="header" class="header"> Dies ist eine Überschrift
    </div>
  </body>
<script>
  // Layer mit Datenquelle
  var osmLayer = new ol.layer.Tile({
    source: new ol.source.OSM()
  });

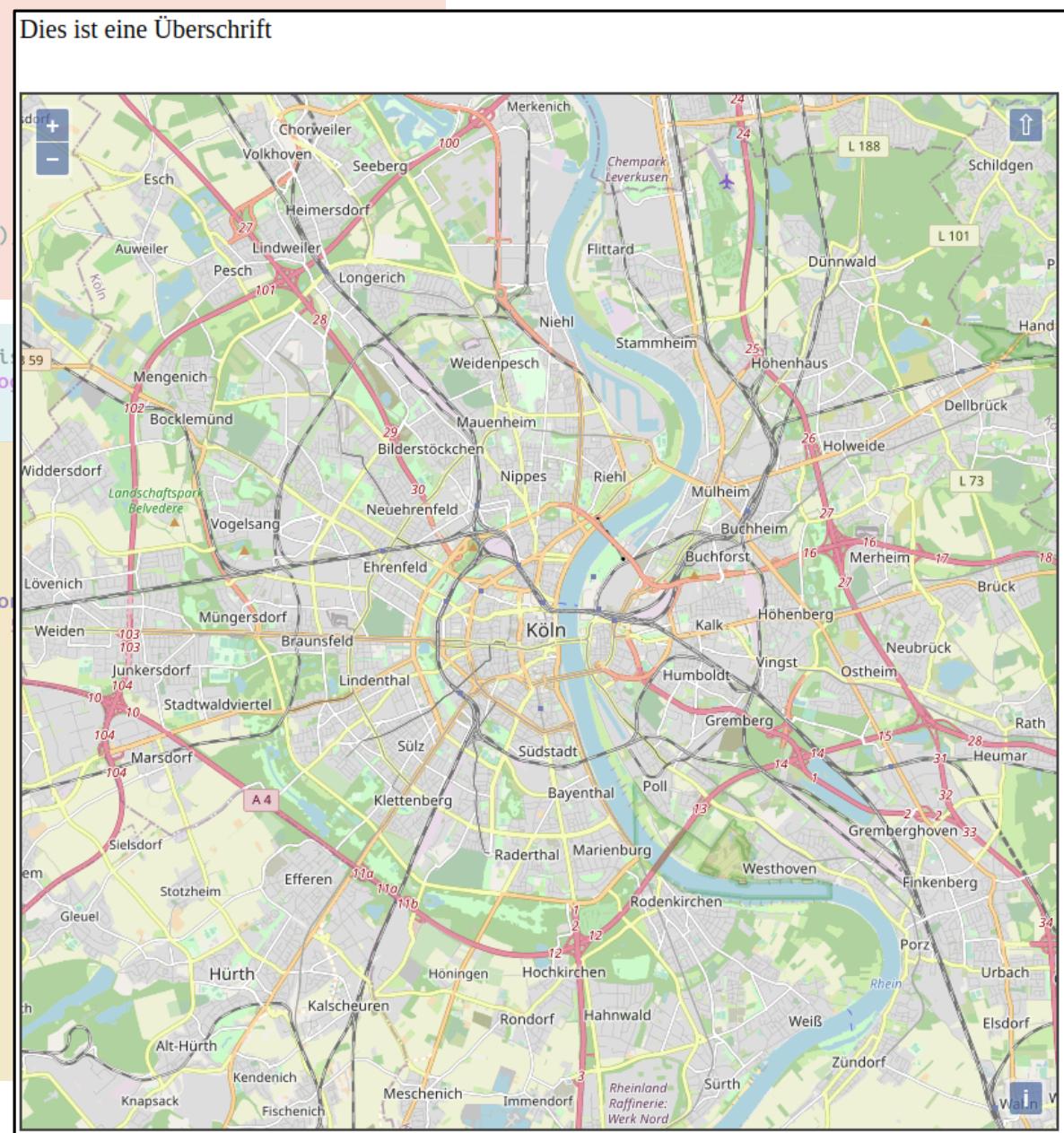
  // Kartenmitte: Kovertierung in Bezugssystem von
  var centerPoint = ol.proj.transform([6.958281,
  50.833333], 'EPSG:4326', 'EPSG:3857');

  // Festlegen der Kartenansicht
  var view = new ol.View({
    center: centerPoint,
    zoom: 12,
    projection: 'EPSG:3857',
    maxZoom: 16,
    minZoom: 4,
  });

  // Kartenerstellung und Verknüpfung zu HTML
  var map = new ol.Map({
    view: view,
    layers: [osmLayer],
    target: 'map'
  });

</script>
</html>

```



Aufbau <head>

Globales Einbinden:

```
<script src="https://cdn.jsdelivr.net/npm/ol@v7.3.0/dist/ol.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/ol@v7.3.0/ol.css">
```

```
<head>
  <title>Beispielkarte Koeln</title>
  <link rel="stylesheet" href="ol.css" type="text/css">
  <script language="javascript" Type="text/javascript"
    src="ol.js"></script>
  <style>
    .header {
      width: 800px;
      height: 50px;
      font-size: 20px;
    }
    .map {
      margin-top: 10px;
      width: 800px;
      height: 800px;
      border: 2px solid rgb(69,69,69);
      background-color: #cccccc;
    }
  </style>
</head>
```

Einbinden des OpenLayer-Stylesheets und der -Bibliothek

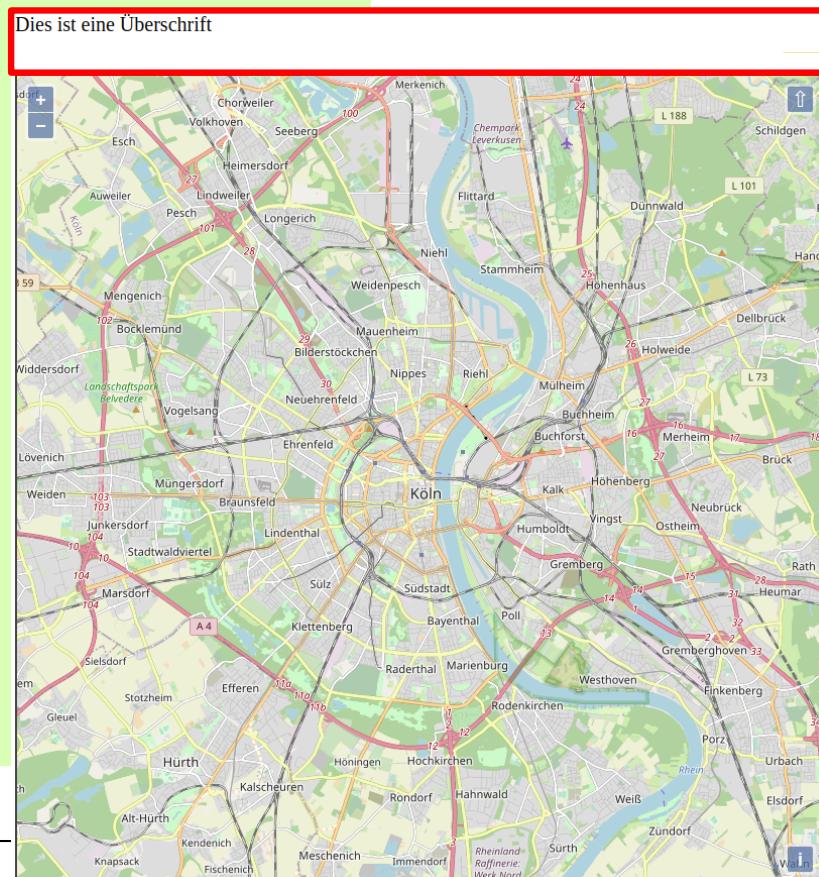
Aufbau <head>

```
<head>
  <title>Beispielkarte Koeln</title>
  <link rel="stylesheet" href="ol.css" type="text/css">
  <script language="javascript" Type="text/javascript"
    src="ol.js"></script>
  <style>
    .header {
      width: 800px;
      height: 50px;
      font-size: 20px;
    }
    .map {
      margin-top: 10px;
      width: 800px;
      height: 800px;
      border: 2px solid rgb(69,69,69);
      background-color: #cccccc;
    }
  </style>
</head>
```

style-Element enthält
Festlegungen der
Gestaltung des HTML
Dokuments

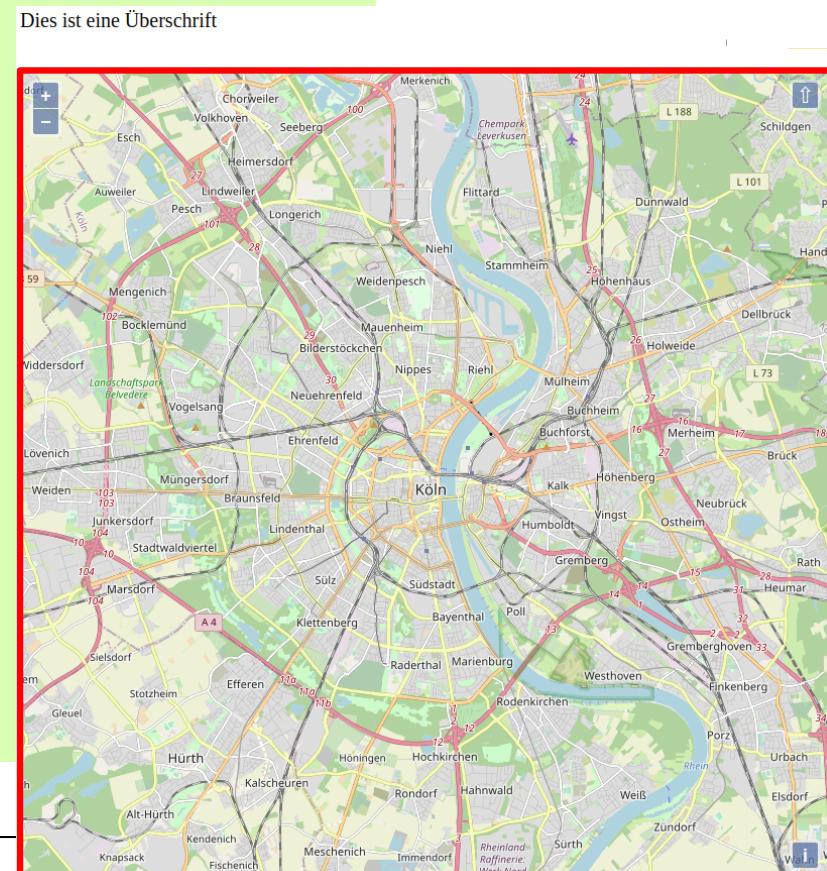
Aufbau <head>

```
<head>
  <title>Beispielkarte Koeln</title>
  <link rel="stylesheet" href="ol.css" type="text/css">
  <script language="javascript" Type="text/javascript"
    src="ol.js"></script>
  <style>
    .header {
      width: 800px;
      height: 50px;
      font-size: 20px;
    }
    .map {
      margin-top: 10px;
      width: 800px;
      height: 800px;
      border: 2px solid rgb(69,69,69);
      background-color: #cccccc;
    }
  </style>
</head>
```



Aufbau <head>

```
<head>
  <title>Beispielkarte Koeln</title>
  <link rel="stylesheet" href="ol.css" type="text/css">
  <script language="javascript" Type="text/javascript"
    src="ol.js"></script>
  <style>
    .header {
      width: 800px;
      height: 50px;
      font-size: 20px;
    }
    .map {
      margin-top: 10px;
      width: 800px;
      height: 800px;
      border: 2px solid rgb(69,69,69);
      background-color: #cccccc;
    }
  </style>
</head>
```



Aufbau <body>

```
<body>
  <div id="header" class="header"> Dies ist eine Überschrift
    
  </div>
  <div id="overlay" class="overlay"></div>
  <div id="map" class="map"></div>
</body>
```

body-Element enthält den
darzustellenden Inhalt eines
HTML-Dokuments

Aufbau <body>

```
<body>
  <div id="header" class="header"> Dies ist eine Überschrift
    
  </div>
  <div id="overlay" class="overlay"></div>
  <div id="map" class="map"></div>
</body>
```

Hier:

- ▷ **header** wird direkt definiert

Aufbau <body>

```
<body>
  <div id="header" class="header"> Dies ist eine Überschrift
    
  </div>
  <div id="overlay" class="overlay"></div>
  <div id="map" class="map"></div>
</body>
```

Hier:

- ▷ **header** wird direkt definiert
- ▷ **overlay**- und **map**-Objekt werden später im **script** über entsprechenden **id**-Tag gefüllt

Aufbau <script>

```
<script>
  // Layer mit Datenquelle
  var osmLayer= new ol.layer.Tile({
    source: new ol.source.OSM()
  });
  var stamenTerrainLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "terrain"})
  );
  var stamenWatercolorLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "watercolor"})
  );
  ...
</script>
```

script-Element enthält alle
Anweisungen in JavaScript

Aufbau <script>

```
<script>
  // Layer mit Datenquelle
  var osmLayer= new ol.layer.Tile({
    source: new ol.source.OSM()
  });
  var stamenTerrainLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "terrain"})
  );
  var stamenWatercolorLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "watercolor"})
  );
...
```

```
</script>
```

Einladen verschiedener Kartenlayer,
sogenannter *Kacheln* (tiles), mittels
OpenLayers

Aufbau <script>

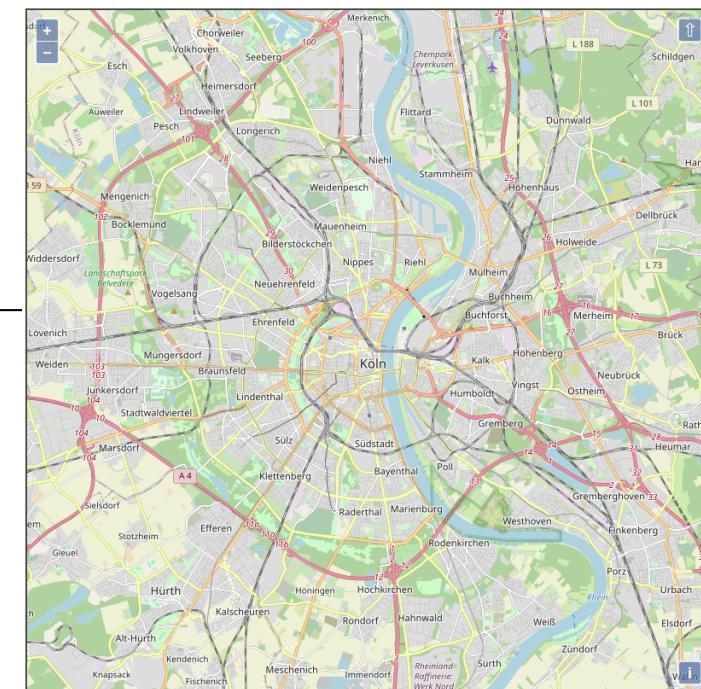
```
<script>
  // Layer mit Datenquelle
  var osmLayer= new ol.layer.Tile({
    source: new ol.source.OSM()
  });

  var stamenTerrainLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "terrain" })
  });

  var stamenWatercolorLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "watercolor" })
  });

  ...

</script>
```



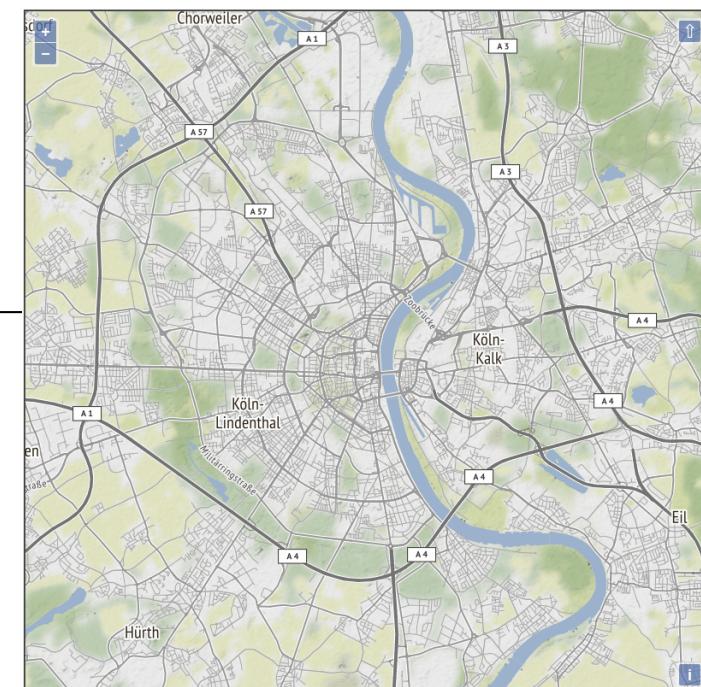
Aufbau <script>

```
<script>
  // Layer mit Datenquelle
  var osmLayer= new ol.layer.Tile({
    source: new ol.source.OSM()
  });

  var stamenTerrainLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "terrain" })
  });

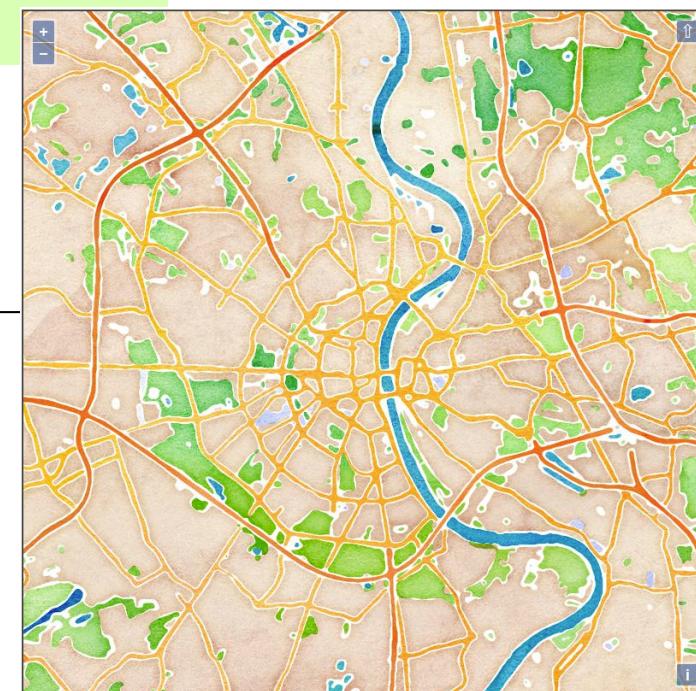
  var stamenWatercolorLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "watercolor" })
  });

  ...
</script>
```



Aufbau <script>

```
<script>
  // Layer mit Datenquelle
  var osmLayer= new ol.layer.Tile({
    source: new ol.source.OSM()
  });
  var stamenTerrainLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "terrain"})
  });
  var stamenWatercolorLayer = new ol.layer.Tile({
    source: new ol.source.Stamen({layer: "watercolor"})
  });
...
</script>
```



Aufbau <script>

```
<script>
...
// Stadtteile Hamburg
var stadtteile = new ol.layer.Vector({
  source: new ol.source.Vector({
    url: "http://geo.local.hcuuh.de/data/districts.geojson",
    format: new ol.format.GeoJSON()
  }),
  maxResolution: 1000,
  minResolution: 5,
  opacity: 1
});
...
</script>
```

Einladen der Stadtteile als selbsterstellte
Vektordaten (GeoJSON), mittels
OpenLayers

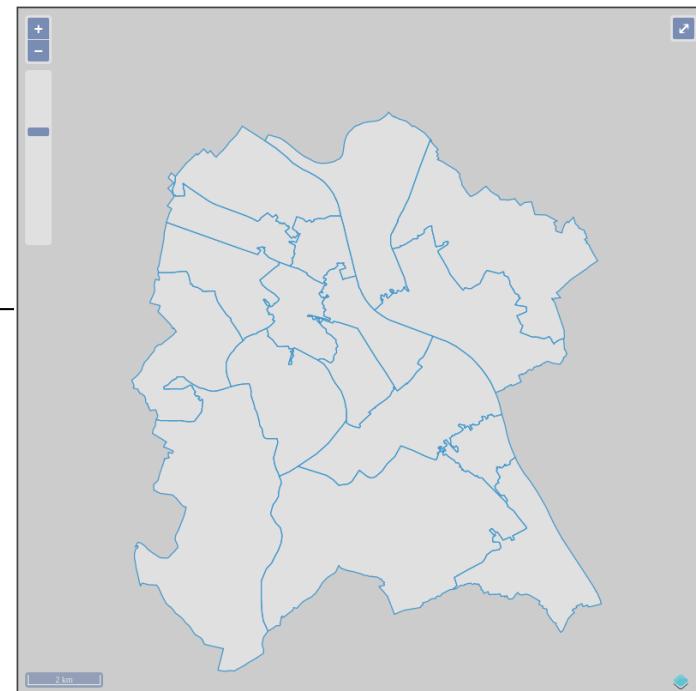
Aufbau <script>

```
<script>
...
// Stadtteile Hamburg
var stadtteile = new ol.layer.Vector({
  source: new ol.source.Vector({
    url: "http://geo.local.hcuhh.de/data/districts.geojson",
    format: new ol.format.GeoJSON()
  }),
  maxResolution: 1000,
  minResolution: 5,
  opacity: 1
});
...
</script>
```

Adresse

Format

sichtbarer Bereich beim
Hinein-/Hinauszoomen der Karte



Aufbau <script>

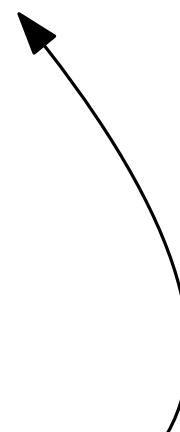
```
<script>
```

```
...
```

```
// Koordinaten der Kartenmitte und  
// Konvertierung WGS84 in Pseudo-Mercator (Bezugssystem von OSM)  
var centerPoint = ol.proj.transform([9.990709,  
    53.531818], "EPSG:4326", "EPSG:3857");
```

```
// Kontrollelemente fuer die Karte  
var controls = [  
    new ol.control.Attribution({collapsed: true}),  
    new ol.control.Zoom(),  
    new ol.control.ScaleLine(),  
    new ol.control.FullScreen(),  
    new ol.control.ZoomSlider()  
];  
...
```

```
</script>
```



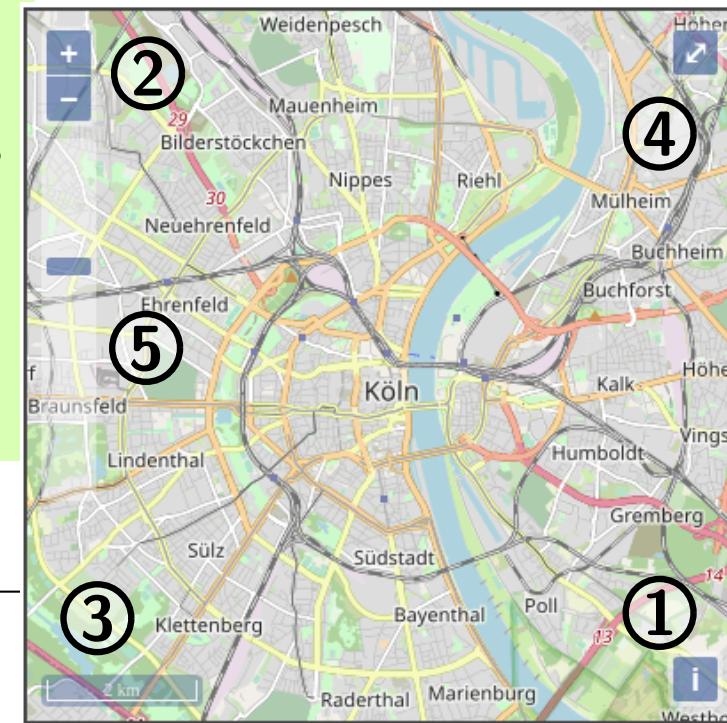
Festlegen der Koordinaten, auf die
die Karte zentriert ist inkl.
Umrechnung, da Hintergrundkarten
in Pseudo-Mercator gegeben

Aufbau <script>

```
<script>
...
// Koordinaten der Kartenmitte und
// Konvertierung WGS84 in Pseudo-Mercator (Bezugssystem von OSM)
var centerPoint = ol.proj.transform([9.990709,
  53.531818], "EPSG:4326", "EPSG:3857");
```

```
// Kontrollelemente fuer die Karte
var controls = [
  1 new ol.control.Attribution({collapsed: true}),
  2 new ol.control.Zoom(),
  3 new ol.control.ScaleLine(),
  4 new ol.control.FullScreen(),
  5 new ol.control.ZoomSlider()
];
...
</script>
```

optionale Kartenelemente



Aufbau <script>

```
<script>
...
// Festlegung der Kartenansicht
var view = new ol.View({
  center: centerPoint,
  zoom: 10,
  projection: "EPSG:3857",
  maxZoom: 18,
  minZoom: 4
});

// Kartenerstellung + Verknuepfung mit HTML (id='map')
var map = new ol.Map({
  view: view,
  layers: [osmLayer],
  controls: controls,
  target: "map"
});
...
</script>
```

Kartenausschnitt wird definiert mit

- ▷ center: Zentrum der Karte
- ▷ zoom: Zoomstufe zu Beginn
- ▷ projection: Referenzsystem
- ▷ maxZoom: maximale Zoomstufe
- ▷ minZoom: minimale Zoomstufe

Aufbau <script>

```
<script>
...
// Festlegung der Kartenansicht
var view = new ol.View({
  center: centerPoint,
  zoom: 10,
  projection: "EPSG:3857",
  maxZoom: 18,
  minZoom: 4
});
```

```
// Kartenerstellung + Verknuepfung mit HTML (id='map')
var map = new ol.Map({
  view: view,
  layers: [osmLayer],
  controls: controls,
  target: "map"
});
```

```
...
</script>
```

Karte wird erzeugt mit

- ▷ view: Kartenausschnitt
- ▷ layers: dargestellte Kartenlayer
- ▷ controls: Kontrollelemente
- ▷ target: Verknüpfung zum HTML-Objekt

Aufbau <script>

```
<script>
...
// Overlay der Postleitzahlen
var overlay = new ol.Overlay({
  element: document.getElementById("overlay"),
  positioning: "bottom-center"
});

map.on("click", function(event) {
  var coor = event.coordinate;
  var text = map.forEachFeatureAtPixel(event.pixel, function(feature) {
    var feature = feature.get("plz");
    return feature;
  });
  var element = overlay.getElement();
  element.innerHTML = text;
  overlay.setPosition(coor);
  map.addOverlay(overlay);
});
</script>
```

Erzeugen eines Overlay-Objektes mit Verknüpfung zum HTML-Objekt (id)

Aufbau <script>

```
<script>
...
// Overlay der Stadtteilen
var overlay = new ol.Overlay({
  element: document.getElementById("overlay"),
  positioning: "bottom-center"
});

map.on("click", function(event) {
  var coor = event.coordinate;
  var text = map.forEachFeatureAtPixel(event.pixel, function(feature) {
    var feature = feature.get("stadtteil_");
    return feature;
  });
  var element = overlay.getElement();
  element.innerHTML = text;
  overlay.setPosition(coor);
  map.addOverlay(overlay);
});
</script>
```

Erzeugen eines
OnClick-Events und
Verknüpfung zum
Overlay-Objekt

Aufbau <script>

```
<script>
...
// Overlay der Stadtteilen
var overlay = new ol.Overlay({
  element: document.getElementById("overlay"),
  positioning: "bottom-center"
});

map.on("click", function(event) {
  var coor = event.coordinate;
  var text = map.forEachFeatureAtPixel(event.pixel, function(feature) {
    var feature = feature.get("stadtteil_");
    return feature;
  });
  var element = overlay.getElement();
  element.innerHTML = text;
  overlay.setPosition(coor);
  map.addOverlay(overlay);
});
</script>
```

Erzeugen eines
OnClick-Events und
Verknüpfung zum
Overlay-Objekt

Position des Mausklick

Aufbau <script>

```
<script>
...
// Overlay der Stadtteilen
var overlay = new ol.Overlay({
  element: document.getElementById("overlay"),
  positioning: "bottom-center"
});

map.on("click", function(event) {
  var coor = event.coordinate;
  var text = map.forEachFeatureAtPixel(event.pixel, function(feature) {
    var feature = feature.get("stadtteil_");
    return feature;
  });
  var element = overlay.getElement();
  element.innerHTML = text;
  overlay.setPosition(coor);
  map.addOverlay(overlay);
});
</script>
```

Erzeugen eines
OnClick-Events und
Verknüpfung zum
Overlay-Objekt

Angezeigter Text:
Stadtteilname des
angeklickten Polygons

Aufbau <script>

```
<script>
...
// Overlay der Stadtteilen
var overlay = new ol.Overlay({
  element: document.getElementById("overlay"),
  positioning: "bottom-center"
});

map.on("click", function(event) {
  var coor = event.coordinate;
  var text = map.forEachFeatureAtPixel(event.pixel, function(feature) {
    var feature = feature.get("stadtteil_");
    return feature;
  });
  var element = overlay.getElement();
  element.innerHTML = text;
  overlay.setPosition(coor);
  map.addOverlay(overlay);
});
</script>
```

Erzeugen eines
OnClick-Events und
Verknüpfung zum
Overlay-Objekt

Verknüpfung vom
HTML-Objekt 'overlay'
mit dem darzustellenden
Text

Aufbau <script>

```
<script>
...
// Overlay der Stadtteilen
var overlay = new ol.Overlay({
  element: document.getElementById("overlay"),
  positioning: "bottom-center"
});

map.on("click", function(event) {
  var coor = event.coordinate;
  var text = map.forEachFeatureAtPixel(event.pixel, function(feature) {
    var feature = feature.get("stadtteil_");
    return feature;
  });
  var element = overlay.getElement();
  element.innerHTML = text;
  overlay.setPosition(coor);
  map.addOverlay(overlay);
});
</script>
```

Erzeugen eines
OnClick-Events und
Verknüpfung zum
Overlay-Objekt

Positionierung des Textes
und Hinzufügen zur Karte

Aufgabe

- ▷ Erstellen Sie eine Webkarte im HTML-Format, welche die Stadtteile in Hamburg visualisiert.
- ▷ Binden Sie neben einer geeigneten Hintergrundkarte die zur Verfügung gestellte GeoJSON-Datei der Stadtteile ein.
 - <http://geo.local.hcuuh.de/data/districts.geojson>
- ▷ Bei einem Mausklick auf die Karte soll der Stadtteilname an dieser Stelle angezeigt werden.
- ▷ Achten Sie auf eine geeignete Wahl des Kartenausschnitts und binden Sie Kontrollelemente ein, die die Navigation innerhalb der Karte erleichtern
- ▷ Für eine ansprechende und vollständige Webvisualisierung fügen Sie ggf. zusätzliche HTML-Objekte ein (<div>-Tags), in welchen Sie zum Beispiel Kartenersteller*in und Erstellungsdatum angeben.