

Essentials of Web Mapping based on JavaScript Libraries

*Munich Technical University //
Chair of Geoinformatics
Murat Kendir, Prof. Thomas
Kolbe // murat.kendir@tum.de*



- In this document we will discuss concepts related to web mapping based on JavaScript libraries.
- Since all web mapping libraries used in this document are based on the JavaScript language, we will also partially discuss some basic concepts of JavaScript.

1) How to combine different sources in HTML?

- In an HTML document, you can combine different resources for use in the web interface.
- Traditionally, files related to style, content and programs are stored in separate files and combined into one HTML file. One popular method of file separation is to use HTML (content), CSS (style) and Javascript (program) files.
- It should be noted that separation is not always the best method. Sometimes developers prefer to use them in a single HTML file.

1.a-b) Defining styles for HTML elements

Demo 1a

```
<title>My Web Page</title>
<div align="center">
  <h1>Welcome to my website!</h1>
  <p align="right">This is my first web page
design.</p>
  <p align="left">I am still very excited to
make this.</p>
</div>
```

Demo 1b

```
<title>My Web Page</title>
<style>
  h1 {text-align:center;}
  p {text-align:left;}
</style>
<div>
  <h1>Welcome to my website!</h1>
  <p>This is my first web page design.</p>
  <p>I am still very excited to make this.</p>
</div>
```

1.c) Referencing to HTML elements

Demo 1c

```
<title>My Web Page</title>
<style>
  #title1 {text-align:center;}
  #paragraph1 {text-align:left;}
  .right {text-align:right;}
</style>
<div>
  <h1 id="title1">Welcome to my website!</h1>
  <p class="right">This is my first web page
design.</p>
  <p id="paragraph1">I am still very excited
to make this.</p>
</div>
```

1.d) Referencing to HTML elements

Demo 1d

```
<title>My Web Page</title>
<style>
  #title1 {text-align:center;}
  #paragraph1 {text-align:left;}
  .right {text-align:right;}
</style>
<div>
  <h1 id="title1">Welcome to my website!</h1>
  <p class="right">This is my first web page
design.</p>
  <p id="paragraph1">I am still very excited
to make this.</p>
  <p id="paragraph1" class="right">What an
exciting moment.</p>
</div>
```

1.e) Importing Styles from an external file

Demo 1e

```
<title>My Web Page</title>

<link rel="stylesheet" href="my_style.css">

<div>
    <h1 id="title1">Welcome to my website!</h1>
    <p class="right">This is my first web page
design.</p>
    <p id="paragraph1">I am still very excited
to make this.</p>
    <p id="paragraph1" class="right">What an
exciting moment.</p>
</div>
```

1.f) Importing Styles from an external file

Demo 1f

```
<title>My Web Page</title>

<link rel="stylesheet" href="https://
www.neredekaliyor.com/koordinat_paylasma/
my_style.css">

<div>
    <h1 id="title1">Welcome to my website!</h1>
    <p class="right">This is my first web page
design.</p>
    <p id="paragraph1">I am still very excited
to make this.</p>
    <p id="paragraph1" class="right">What an
exciting moment.</p>
</div>
```

2.a) How to configure modern web sites using JavaScript libraries?

- Similar to CSS implementations, JavaScript code can be applied to HTML files using internal or external sources.
- Modern websites often combine multiple JavaScript libraries.

An Example: **ThreeJS**

- Not only for geospatial use cases, but also animations, 3D models etc.
- is base library of various geospatial JavaScript libraries (e.g. **three-geo**, **three-globe**)
- Implemented into QGIS to use as 3D Visualizer
 - accessible through **View > 3D Map Views**

Demo 2a

3.a) Web Map with OpenLayers

- OpenLayers is one of the most advanced 2D web map libraries.

Demo3a

```
<html>
  <head>
    <script src="https://cdn.jsdelivr.net/npm/
ol@v7.2.2/dist/ol.js"></script>
    <link rel="stylesheet" href="https://
cdn.jsdelivr.net/npm/ol@v7.2.2/ol.css">
    <title>Empty Map with defaults</title>
    <link rel="stylesheet" href="demo3a.css"></
link>
  </head>
  <body>
    <div id="map"></div>
    <div class="box"> Check the style of this
box. </div>
    <script type="text/javascript"
src="demo3a.js"></script>
  </body>
</html>
```

3.b) Understanding the JavaScript code

```
var map = new ol.Map({
  target: 'map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()})
  ],
  view: new ol.View({
    center: [1287392, 6123424],
    zoom: 10
  })
});
```

- var map --> defines a custom variable
- new ol.Map --> calls the **Map** class from the **ol** module.
 - target: 'map' --> sets target HTML element as "map"
 - layers: [] --> adds some layer to show in map object
 - new ol.layer.Tile() --> adds a layer as a tiled map
 - new ol.source.OSM() --> sets the source of the tile map as

OSM

- view: `new ol.View()` --> sets view as a new **View** object.

3.c) Finding the elements of a JavaScript object

- Type `console.dir(map)` in the (Web) Developer Tool of browser
- After finding the target element, you can print the content with `console.log(map.targetElement_)`
- Search for the target, layers and view elements.

```

▶ pixelToCoordinateTransform_: Array(6) [ 47.291789398002896, 0, 0, ... ]
▶ postRenderFunctions_: Array []
  postRenderTimeoutHandle_: undefined
▶ previousExtent_: Array(4) [ 3577758.9955922775, 4829559.231667059, 3695704.7183510466, ... ]
  renderComplete_: undefined
▶ renderer_: Object { disposed: false, map_: {...}, renderedVisible_: true, ... }
▶ resizeObserver_: ResizeObserver { }
  revision_: 0
▶ targetChangeHandlerKeys_: Array [ {...}, {...} ]
▼ targetElement_: <div id="map">
  accessKey: ""
  accessKeyLabel: ""
  align: ""
  ariaAtomic: null
  ariaAutoComplete: null
  ariaBrailleLabel: null

```

3.c) Finding the elements of a JavaScript object

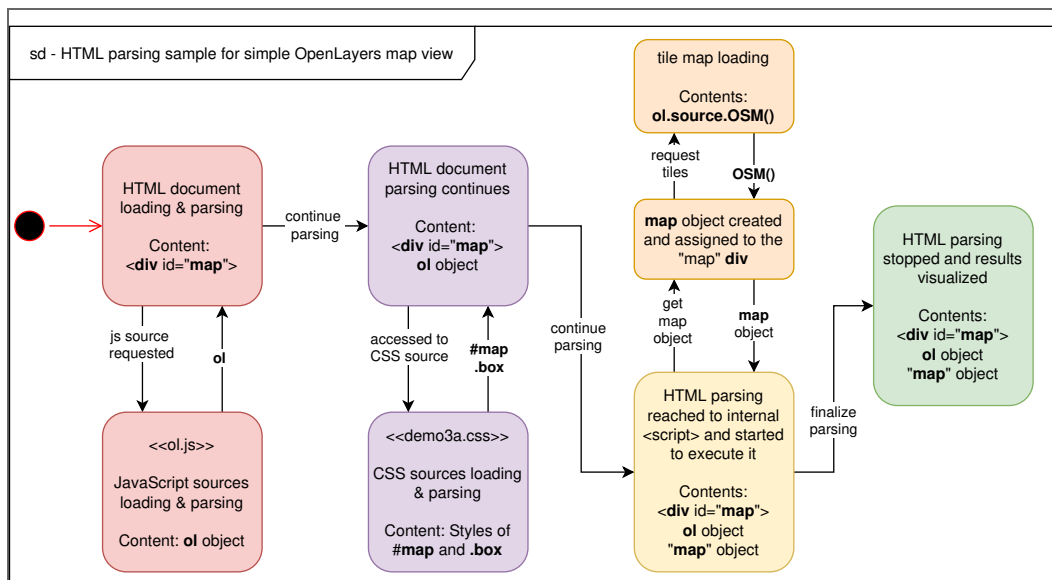
- Try to find the specified map center coordinates `console.dir(map)`

- Check

`console.log(map.viewPropertyListe`

```
▶ pendingRemovals_: Object { ... }
▶ projection_: Object { code_: "EPSG:3857", units_: "m", axisOrientation_: "enu", ... }
  resolutions_: undefined
  revision_: 16
▶ targetCenter_: Array [ 1289376.6454045859, 6127338.667012726 ]
  targetResolution_: 82.10083050882518
  targetRotation_: 0
▶ un: function unInternal(t, e) { ... }
  updateAnimationKey_: undefined
▼ values_: Object { zoom: 10.896874874409963, rotation: 0, resolution: 82.10083050882518, ... }
  ▶ center: Array [ 1289376.6454045859, 6127338.667012726 ]
    resolution: 82.10083050882518
    rotation: 0
    zoom: 10.896874874409963
  ▶ <prototype>: Object { ... }
▶ viewportSize_: Array [ 2494, 294 ]
  zoomFactor_: 2
▶ <prototype>: Object { ... }
```

3.d) State Diagram for a HTML Parsing Process (HTML+CSS+OpenLayers)



4.a) In which Coordinate Reference System (CRS)?

- Even if a custom CRS is not defined, the map has to be visualized based on a default CRS.

- Check

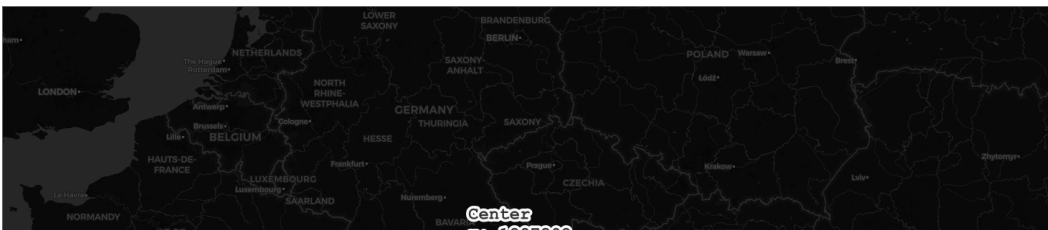
```
console.log(map.viewPropertyListe
```

- Search with the EPSG code on epsg.io to learn more about it.

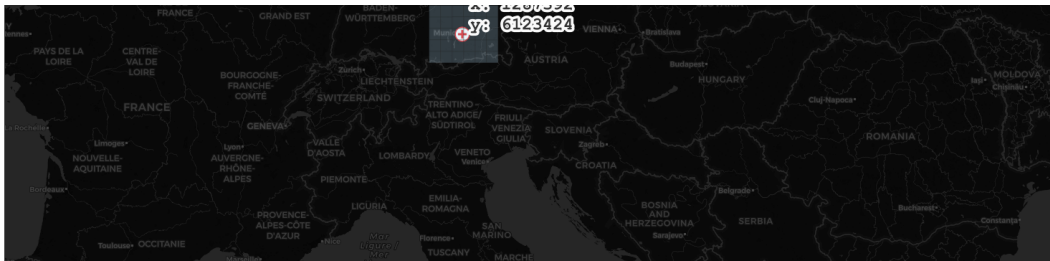
```
>> console.log(map.viewPropertyListenerKey_.target.projection_)
Object { code_: "EPSG:3857", units_: "m", extent_: (4) [...], worldExtent_: (4) [...], axisOrientation_: "enu", global_: true, defaultTileGrid_: null, metersPerUnit_: undefined }
  axisOrientation_: "enu"
  canWrapX_: true
  code_: "EPSG:3857"
  defaultTileGrid_: null
  extent_: Array(4) [ -20037508.342789244, -20037508.342789244, 20037508.342789244, ... ]
  getPointResolutionFunc_: function getPointResolution(t, e) {
    global_: true
    metersPerUnit_: undefined
    units_: "m"
  }
  worldExtent_: Array(4) [ -180, -85, 180, ... ]
  <prototype>: Object { ... }
← undefined
```

4.b) How about the Zoom Level?

- The zoom level is defined as 10 in the JavaScript code.
- Many web map applications are using the zoom level as one of the main map constructor.
 - e.g. [https://
www.openstreetmap.org/
#map=10/48.1038/11.6483](https://www.openstreetmap.org/#map=10/48.1038/11.6483)
- To learn and test the zoom levels in the same CRS, you can use some plugins in QGIS:
 - Set the CRS as **EPSG:3857 (Pseudo-Mercator)** in Map Properties
 - Search for "**Zoom Level**" and/or "**Layer Tiles Mapcanvas**" in Plugins Manager



Spatial Data Infrastructures (SDI) slides



4.c) Get the current zoom level and write to an HTML element

Demo4c

```
function onZoomEnd(evt) {  
  const map = evt.map;  
  var zoomInfo = 'Zoom level : \n' +  
map.getView().getZoom(); ;  
  console.log(zoomInfo);  
  
document.getElementById('topright_box').innerHTM  
L = zoomInfo;  
}  
  
map.on('moveend', onZoomEnd);
```

5.a) How to add WMS service to the OpenLayers?

Which one is a correct definition for a WMS service in OpenLayers?

Demo5a

A

```
new ol.layer.Image({  
  source: new ol.source.ImageWMS({  
    url: 'https://geoportal.muenchen.de/  
geoserver/gsm_wfs/awm_wertstoffhoeffe_opendata/  
ows',  
    params: {'LAYERS':  
'awm_wertstoffhoeffe_opendata'},  
  }),  
});
```

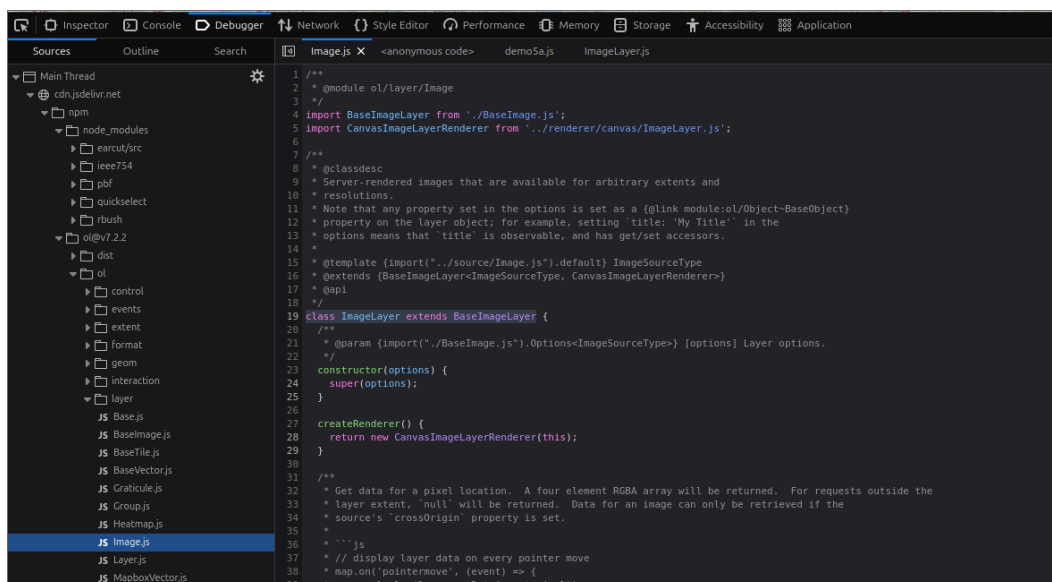
B

```
new Image({  
  source: new ImageWMS({  
    url: 'https://geoportal.muenchen.de/  
geoserver/gsm_wfs/awm_wertstoffhoeffe_opendata/  
ows',  
    params: {'LAYERS':  
'awm_wertstoffhoeffe_opendata'},  
  }),  
});
```

5.b) How to detect the module hierarchy in a package?

Method 1: Check the debugger tool in (Web) Developer Tool.

Method 2: Check the documentation of the used library (e.g. **OpenLayers API Doc**).



5.c) How to get feature info?

Demo5c

- Check the GetFeatureInfo operation's documentation: **OGC e-Learning**
- Check the following classes/methods in OpenLayers API Doc:
 - **Search for "on(type, listener)" method**
 - **Search for "getFeatureInfoUrl" method**

```
map.on('singleclick', function (evt) {  
    //console.log(evt);  
  
    document.getElementById('topright_box').innerHTML = '';  
    const viewResolution = view.getResolution();  
    const url = wmsSource.getFeatureInfoUrl(  
        evt.coordinate,  
        viewResolution,  
        'EPSG:3857',  
        {'INFO_FORMAT': 'text/plain'},  
    );  
});
```

5.d) Display the feature information in a HTML element

Demo5c

- If you are not familiar with some JavaScript functions or classes like 'let', 'const', 'Promise', check the documents for the 2015 JavaScript version (aka ECMAScript 6/ES6):

- **https://www.w3schools.com/Js/js_es6.asp**

```
map.on('singleclick', function (evt) {
    //console.log(evt);

    document.getElementById('topright_box').innerHTML = '';
    const viewResolution = view.getResolution();
    const url = wmsSource.getFeatureInfoUrl(
        evt.coordinate,
        viewResolution,
        'EPSG:3857',
        {'INFO_FORMAT': 'text/plain'},
    );
    if (url) {
        fetch(url)
            .then((response) => response.text())
            .then((html) => {

                document.getElementById('topright_box').innerHTML = html;
            });
    }
});
```

```
}  
});
```

6.a) How to add WMTS layer to the OpenLayers?

USGS WMTS service (Capabilities)

- Which one is a correct definition for a WMTS connection?

A

```
const my_wmts = new ol.layer.Tile({
  opacity: 0.5,
  source: new ol.source.WMTS({
    url: 'https://
mrdata.usgs.gov/mapcache/wmts',
    layer: 'sgmc2',
    width: '256px',
    height: '256px',
    format: 'image/png',
  }),
});
```

B

```
const my_wmts = new ol.layer.Tile({
  opacity: 0.5,
  source: new ol.source.WMTS({
    url: 'https://
mrdata.usgs.gov/mapcache/wmts',
    layer: 'sgmc2',
    matrixSet:
'GoogleMapsCompatible',
    format: 'image/png',
  }),
});
```


6.b) What are the resolution and MatrixSet IDs?

- Also check the EPSG:3857 (Pseudo-Mercator) extents : **EPSG.io**

Demo6a

```
const projection = ol.proj.get('EPSG:3857');
const projectionExtent = projection.getExtent();
console.log('Projection Extent : ',
projectionExtent);
const size =
ol.extent.getWidth(projectionExtent) / 256;
console.log('Size : ' , size);
const resolutions = new Array(19);
const matrixIds = new Array(19);
for (let z = 0; z < 19; ++z) {
  // generate resolutions and matrixIds arrays
  for this WMTS
  resolutions[z] = size / Math.pow(2, z);
  matrixIds[z] = z;
}
```

6.c) Implementing the resolution and MatrixSet IDs with openlayers WMTS request

Demo6a

```
const my_wmts = new ol.layer.Tile({
  opacity: 0.5,
  source: new ol.source.WMTS({
    url: 'https://
mrdata.usgs.gov/mapcache/wmts',
    layer: 'sgmc2',
    matrixSet:
'GoogleMapsCompatible',
    format: 'image/png',
    projection: projection,
    tileGrid: new
ol.tilegrid.WMTS({
      origin:
ol.extent.getTopLeft(projectionExtent),
      resolutions: resolutions,
      matrixIds: matrixIds,
    }),
    style: 'default',
    wrapX: false,
  }),
});
```

7.a) 3D Mapping with Cesium

- The existing base map options in CesiumJS need authorization (API token) to limit map requests (e.g. Bing Map)
- However, OpenStreetMap offers a tile basemap that does not require any authorization and can be used in CesiumJS.

```
<html lang="en">
  <head>
    <title>Cesium Demo</title>
    <script src="../../cesium/Build/Cesium/
Cesium.js"></script>
    <link href="../../cesium/Build/Cesium/Widgets/
widgets.css" rel="stylesheet">
  </head>
  <body style="width:100%;height:100%;">
    <div id="cesiumContainer"></div>
    <script>
      // Initialize the Cesium Viewer in the HTML
      element with the `cesiumContainer` ID.
      const viewer = new
Cesium.Viewer('cesiumContainer', {
        //Use OpenStreetMaps
        imageryProvider : new
Cesium.OpenStreetMapImageryProvider({
          url : 'https://
a.tile.openstreetmap.org/'
        }),
      });
    </script>
```

```
</div>  
</body>  
</html>
```

7b) Converting local JS file references to cloud repository references

Cesium with online repositories

- Usually online JS repositories provided in the documentation page of the library.
 - However, you can also check some known JS repositories such as **cdnjs**, **jsdelivr** etc.

```
<html lang="en">
  <head>
    <title>Cesium Demo</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/cesium/1.124.0/Cesium.js"></script>
    <link href="https://cdn.jsdelivr.net/npm/cesium@1.124.0/Build/Cesium/Widgets/widgets.min.css" rel="stylesheet">
  </head>
  <body style="width:100%;height:100%;">
    <div id="cesiumContainer"></div>
    <script>
      // Initialize the Cesium Viewer in the HTML
      element with the `cesiumContainer` ID.
      const viewer = new
      Cesium.Viewer('cesiumContainer', {
        //Use OpenStreetMaps
        imageryProvider : new
        Cesium.OpenStreetMapImageryProvider({
          url : 'https://
a.tile.openstreetmap.org/'
        }),
      });
    </script>
```

```
</div>  
</body>  
</html>
```

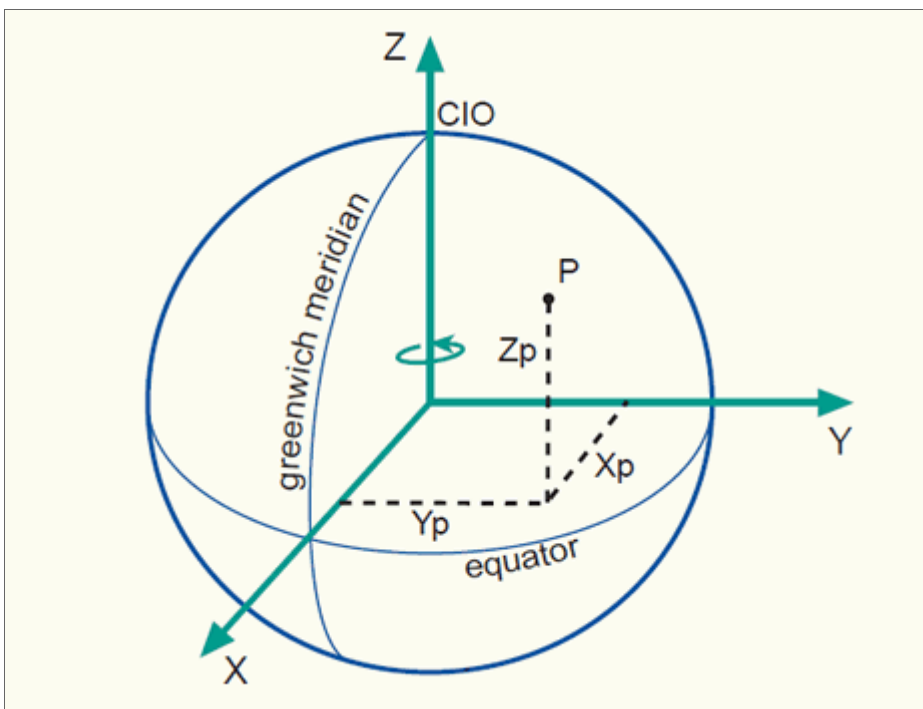
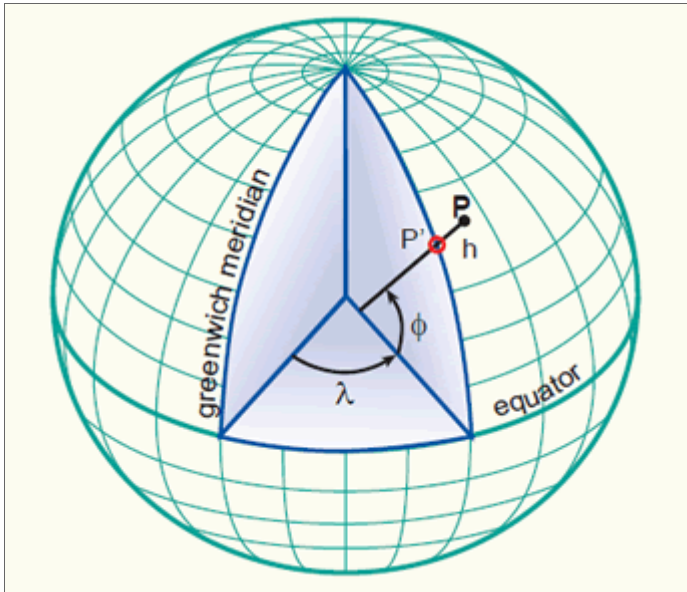
7.c) Camera Object

- Unlike 2D maps, 3D map environments need a camera object so that a scene selected on the map can be presented to the end user.

Cesium with camera initialization

```
// Fly the camera to San Francisco at the given  
longitude, latitude, and height.  
viewer.camera.flyTo({  
  destination :  
Cesium.Cartesian3.fromDegrees(11.575,48.107,  
4000),  
  orientation : {  
    heading : Cesium.Math.toRadians(0.0),  
    pitch : Cesium.Math.toRadians(-45.0),  
  }  
});
```

7.d) What is "Cartesian3"?



- Images taken from <https://kartoweb.itc.nl/geometrics/Coordinate%20systems/coordsys.html>

7.e) How the basemaps are loaded into Cesium (Tiling Mechanism)

- Visit the next demo page and try to give some tilt angle to the current view.
- Unfold the Cesium Viewer Inspector and find the Terrain group at the bottom.
- Enable the "Show Tile Coordinates" and the "Suspend LOD Update" options, then try zooming out.
- This is the basic 2D tiling mechanism that used to overlay 2D maps and terrain files.

Cesium with viewer Inspector

7.f) How to add WMS service to the Cesium

- CesiumJS simply calculates the coverage area and tries to request for the WMS image by comparing with the current view area of the camera (frustum).
- Check the Capabilities of a found WMS service: **München Bezirksteil**

Cesium with WMS service

```
// Add the main URL of WMS service
const geourl = 'https://geoportal.muenchen.de/
geoserver/gsm_wfs/vablock_bezirksteil_opendata/
ows?';
//Define the WMS service using available
parameters
const geoWmsLayer = new
Cesium.WebMapServiceImageryProvider({
  url : geourl,
  parameters: {
    service : 'WMS',
    format : 'image/png',
    transparent : true,
  },
  layers : 'vablock_bezirksteil_opendata',
  EPSG: '4326'
});
//Add to the list of imagery layers of the
current viewer.
viewer.imageryLayers.addImageryProvider(geoWmsLa
yer);
```

7.g) How to add default world terrain using Cesium Ion token

Cesium with Terrain

- Create an account on Cesium Ion and generate a token specifically reserved for an application or limited in time.
- Add to the beginning of your CesiumJS code.

```
//Change the token here below  
Cesium.Ion.defaultAccessToken =  
"xyzabcxyzabcxyzabcxyzabcxyzabcxyzabcxyzabcxyzabcxyzabcxyzabc";
```

- Go to Asset Manager in Cesium Ion web site and find the ID number of the world terrain.
- Then add the terrain by using its ID number as below.

```
viewer.scene.setTerrain(  
    new Cesium.Terrain(  
        Cesium.CesiumTerrainProvider.fromIonAssetId(1),  
    ),  
);
```

8.a) Adding 3D Tiles

Cesium with 3D Tiles

- Add the main JSON link of the 3DTiles to Cesium

- e.g. **BaseMap.de - LOD2**

Buildings

```
var tileset = viewer.scene.primitives.add(new
Cesium.Cesium3DTileset({
    url: 'https://web3d.basemap.de/cesium/
buildings-fly/root.json'
}));
```

- You can add attribute specific styles

```
var cityStyle = new Cesium.Cesium3DTileStyle({
    color : {
        conditions : [
            ["${surface} === 'wall'",
"color('#f2f2f2)"],
            ["${surface} === 'roof'",
"color('#ff5c4d)"],
            ["${surface} === 'bridge'",
"color('#999999)"]
        ]
    },
});
```

8.b) Enable Cesium 3DTiles Inspector

Cesium with 3D Tiles and Inspector

```
viewer.extend(Cesium.viewerCesium3DTilesInspectorMixin);
```

- Try to play with "Dynamic Screen Space Error" values
- Check the "Memory Usage" and the "Rendering Statistics"