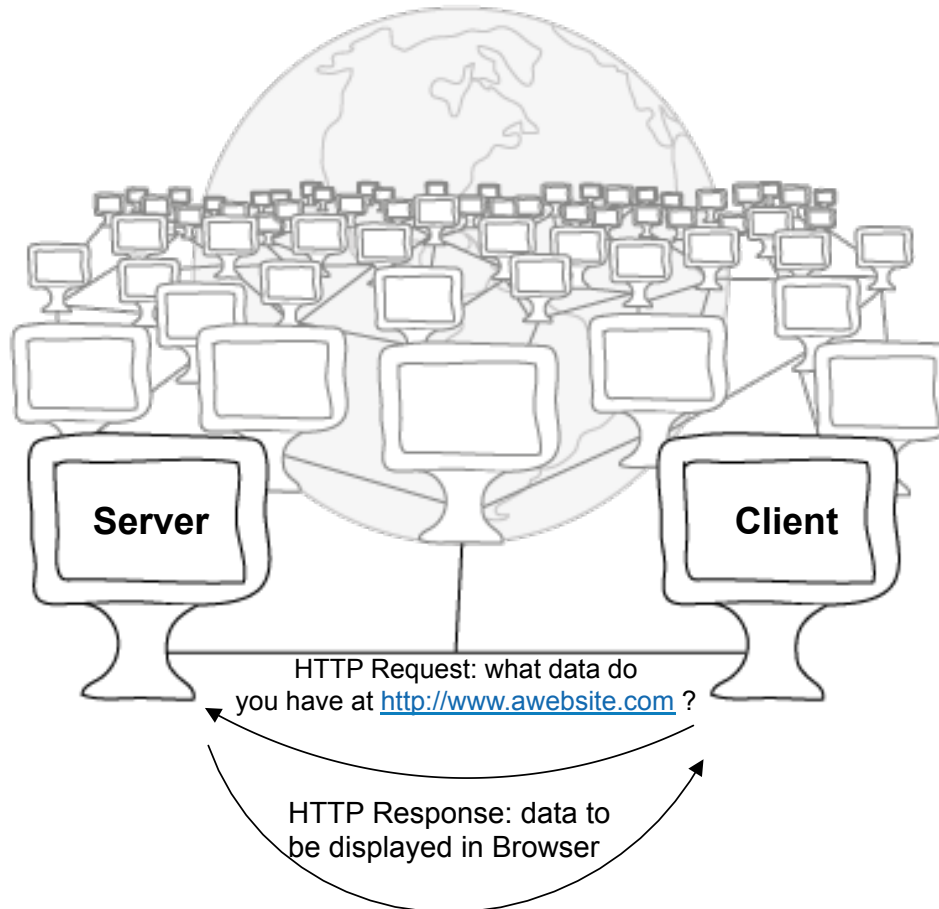




The Web Environment and Web Mapping Libraries

Ionuț Iosifescu

Internet vs. Web

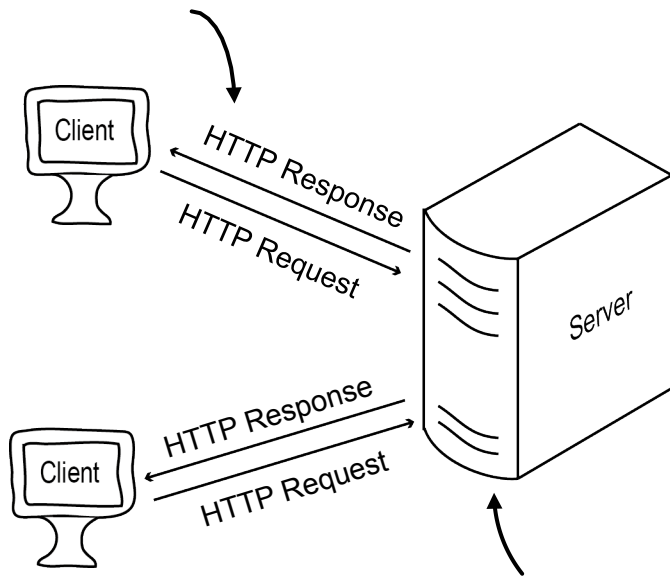


HyperText Transfer Protocol (HTTP) – communication protocol for the web

Web Server

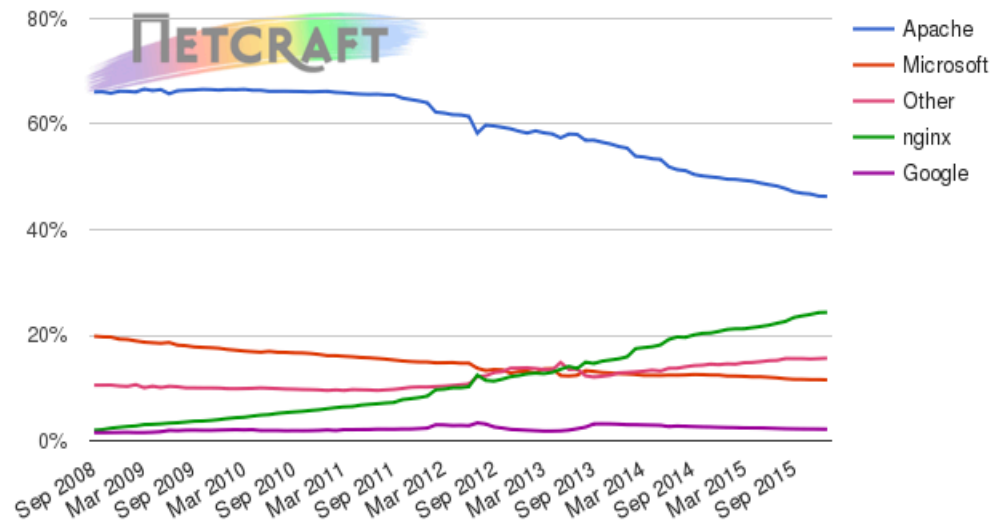
The web content sent in the HTTP response could be:

- **Static:** HTML pages, files available on the server
- **Dynamic:** the content is generated on the fly with each request using: CGI scripts, PHP, Python, JavaServer Pages etc.



On the server there is a Web server installed, that handles the requests

Web server developers: Market share of the top million busiest sites



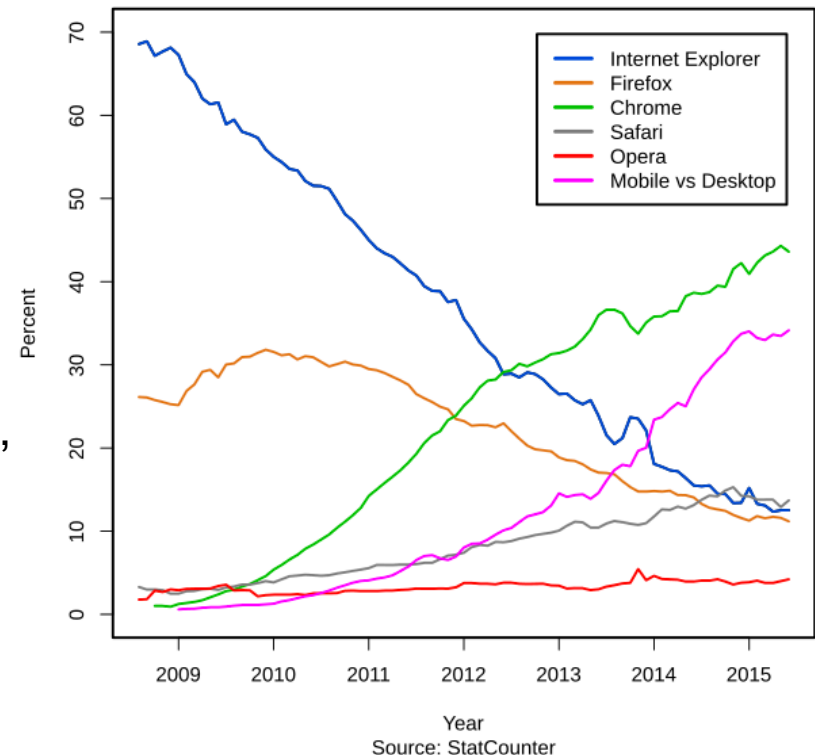
Source: <http://news.netcraft.com/>

Web Client

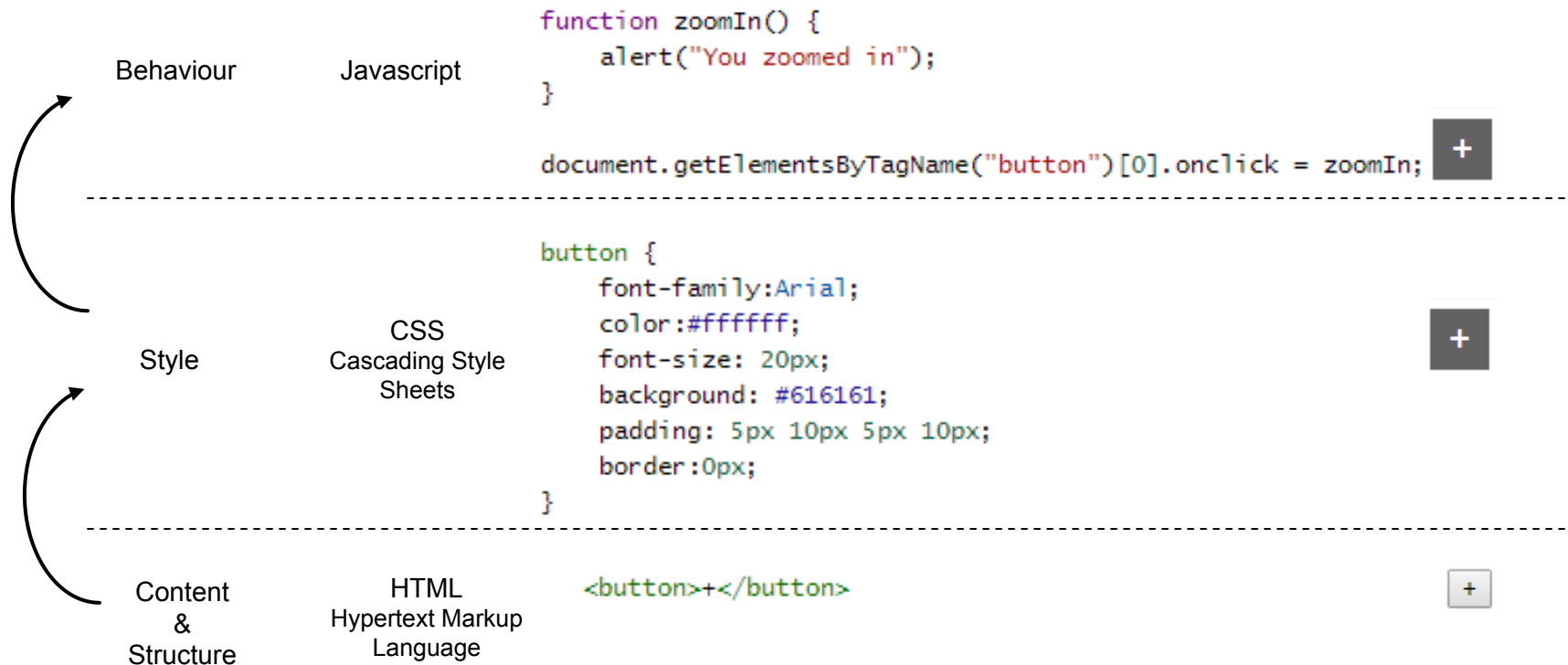
Web Browser

- sends requests to a web server, receives data, interprets and displays them
- standard formats: **HTML**, JPEG, PNG, GIF, SVG
- can natively interpret **CSS** for layout and **Javascript** for user interaction

Usage share of web browsers



Structure, content, style and behaviour



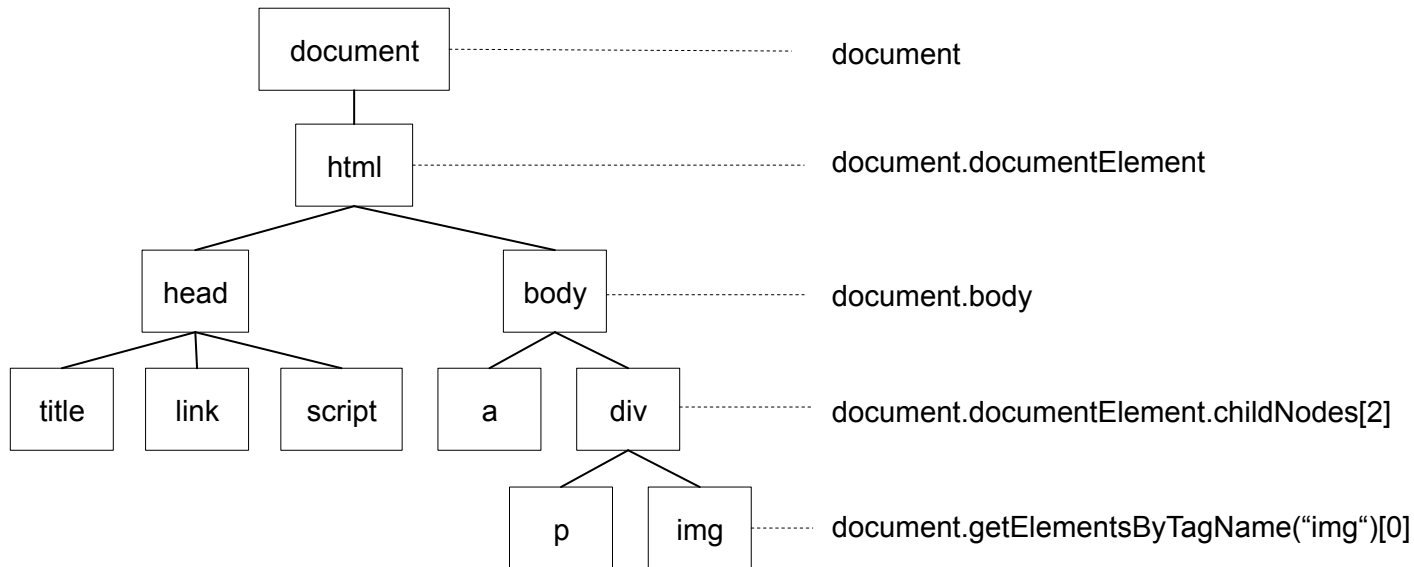
HTML – Hyper Text Markup Language

<tagname>content</tagname>

```
<!DOCTYPE html>
<html>
  <head>
    <title>My first Web Page</title>
    <link rel="stylesheet" type="text/css" href="css/style.css">
    <script src="js/code.js"></script>
  </head>
  <body>
    <div>
      <p>This is my first paragraph</p>
      <a href="http://www.mywebsite.com">Visit my website!</a>
    </div>
    
  </body>
</html>
```

An overview of tags: <http://www.w3schools.com/tags/default.asp>

HTML DOM – Document Object Model



CSS – Cascading Style Sheets

```
selector {  
  property: value;  
  property: value;  
}
```

```
body {  
  background-color: white;  
  color: black;  
}
```

```
p, li {  
  line-height: 12px;  
  color: green;  
}
```

```
<p id= "green" >This text should be green.</p>
```

```
#green {  
  color: green;  
}
```

```
<p class= "red" >This text should be red.</p>
```

```
.red {  
  color: red;  
}
```

```
<p class= "red" >This text should also be red.</p>
```

An overview of CSS properties: <http://www.w3schools.com/cssref/default.asp>

Javascript – variables

var name = value

- Single value variables can store:

- Number `var zoom = 20`
- String `var name = "baseLayer"`
- Boolean `var check = true`

- Arrays can store any number of values

0 1 2 3 4
↓ ↓ ↓ ↓ ↓
`var array = ["citiesLayer" , "baseLayer" , "riversLayer" , 20, true]`

`array[0] = "capitalsLayer"`

↙
`var array = ["capitalsLayer" , "baseLayer" , "riversLayer" , 20, true]`

Javascript – functions

```
function name (parameter1, parameter2) {  
  ...some code here...  
  return value; //optional  
}
```

```
function setColor(polygonPopulation) {  
  if (polygonPopulation<100000) {color = "#ffffcc"}  
    else if (polygonPopulation<500000) {color = "#78c679"}  
    else {color = "#ffffff"}  
  
  return color;  
}  
polygonColor = setColor(polygonPopulation);
```

Javascript - objects

```
var object = {  
    property1: value1,  
    property2: value2,  
}
```

```
var map = {  
    zoom: 10,  
    layers: [],  
    center: {  
        lng: 10.50,  
        lat: 45.50  
    },  
    zoomIn: function() {  
        this.zoom = this.zoom + 1;  
    }  
}
```

```
map.zoom = 9; //will set the map zoom to 9  
map.layers[0] = "baselayer"; //will set the first layer (index 0) to baselayer  
map.center.lng = 11; //will set the longitude of the center to 11  
map.zoomIn(); //will zoom in the map
```

Object	property: value type
map	zoom: number
	layers: array
	center: object {lng, lat}
	zoomIn: function ()

Javascript – event handlers

“Events are sent to notify code of interesting things that have taken place.”

MDN – Mozilla Developer Network

Events:

- user clicks the search button

→ a function that geocodes the location given by the user

```
document.getElementById("searchButton").addEventListener("click", searchLocation);  
function searchLocation() { ...code here... }
```

- user scrolls to zoom in

→ a function that makes the map zoom in

```
document.getElementById("mapDiv").addEventListener("scroll", zoomIn);  
function zoomIn() { ...code here... }
```

- the body finishes loading

→ a function that initializes the map

```
document.body.addEventListener("load", initMap);  
function initMap() { ...code here... }
```

More events here: <https://developer.mozilla.org/en-US/docs/Web/Events>

Web Mapping Libraries

```
/* * L.Map is the central class of the API - it is used to create a map. */
```

```
L.Map = L.Class.extend({
  includes: L.Mixin.Events,
  options: {
```



```
  crs: L.CRS.EPSG3857,
  fadeAnimation: L.DomUtil.TRANSITION && !L.Browser.android23,
  trackResize: true,
  markerZoomAnimation: L.DomUtil.TRANSITION && L.Browser.any3d
```



```
  initialize: function (id, options) { // (HTMLElement or String, Object)
    options = L.setOptions(this, options);
    this._initContainer(id);
    this._initLayout();
    this._onResize = L.bind(this._onResize, this);
```



```
  this._initEvents();
  if (options.maxBounds) {
    this.setMaxBounds(options.maxBounds);
  }
```



```
  if (options.center && options.zoom !== undefined) {
    this.setView(L.latLng(options.center), options.zoom);
```



```
  this._handlers = [];
  this._layers = [];
  this._addLayers(options.layers);
```

```
}, // public methods that modify map state // resetView by animation-powered implementation in Map.PanAnimation.js
```



```
  resetView: function (center, zoom) {
    zoom = zoom === undefined ? this.getZoom() : zoom;
    this.resetView(L.latLng(center), this._limitZoom(zoom));
    return this;
  },
```

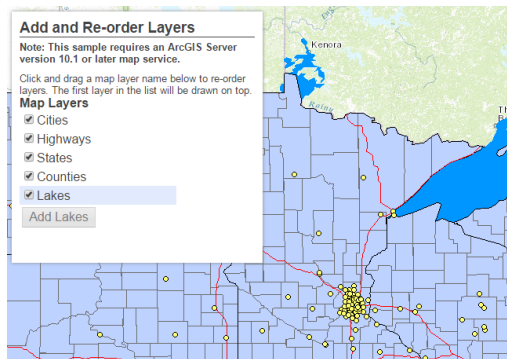


```
  zoom: function (zoom, options) {
    if (!this._loaded) { this._zoom = this._limitZoom(zoom); return this; }
    return this.setView(this.getCenter(), zoom, {zoom: options});
  },
  ...}

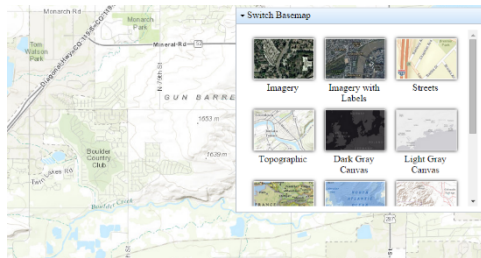
```

What can mapping libraries do?

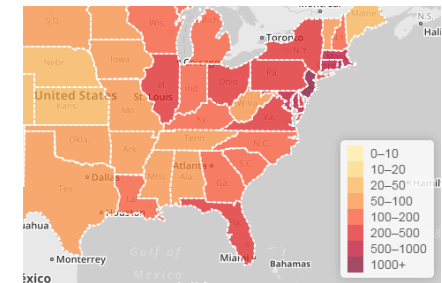
Add layers with your own data



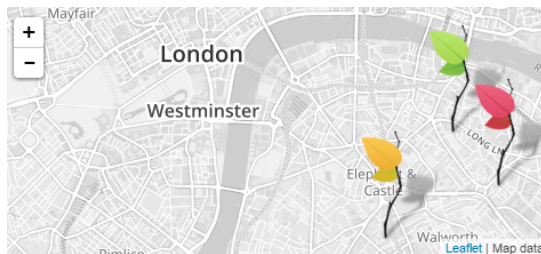
Choose from a variety of basemaps and use it for your map



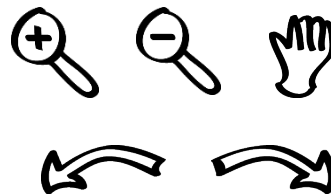
Add a legend



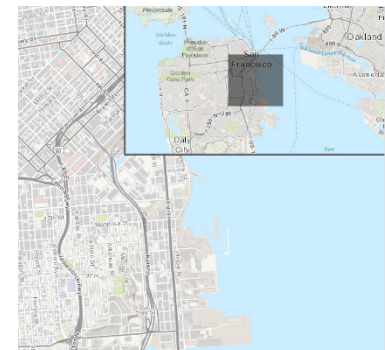
Style your data



Navigate on the map

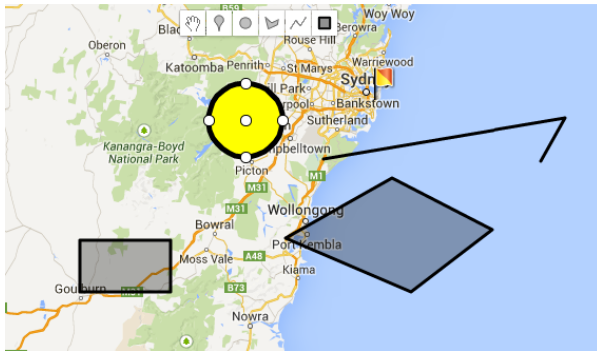


Add an overview map

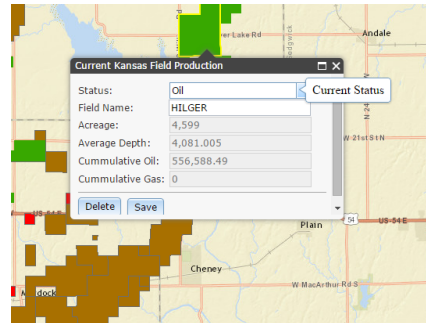


What can mapping libraries do?

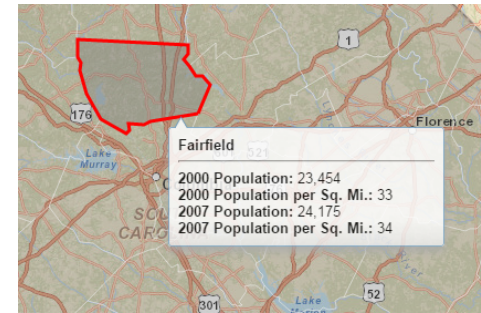
Draw on the map



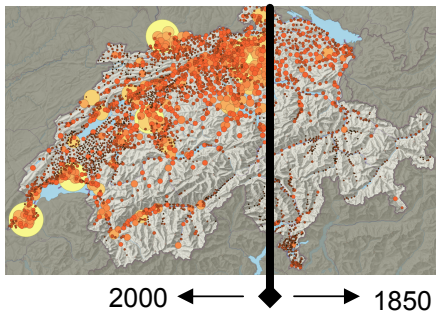
Edit data



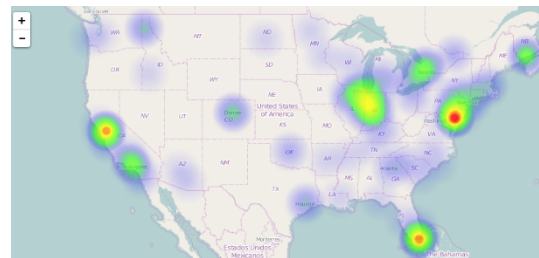
Select and query the data



Compare data with swipe tool



Create a heatmap



- Cluster points
- Geocode addresses
- Create a buffer
- Draw a profile

...

Functionality Leaflet (part I)

Map
L.map()

<http://leafletjs.com/reference.html>

Zoom functionality
L.control.zoom()

Basemap as tileLayer
L.tileLayer()

Scale control
L.control.scale()

Layers Control
L.control.layers()

Load data (polygon, line, point) as GeoJSON
L.geoJson()

Custom Control ex. Legend
L.control()

Renewable energy
Hover over a country

0-10
10-20
20-30
30+

Functionality Leaflet (part II)

Function that
creates the
object

Factory	Description
<code>L.map(<HTMLElement String> id, <Map_options> options?)</code>	Instantiates a map object given a div element (or its id) and optionally an object literal with map options described below.

Properties

Option	Type	Default	Description
center	LatLng	null	Initial geographical center of the map.

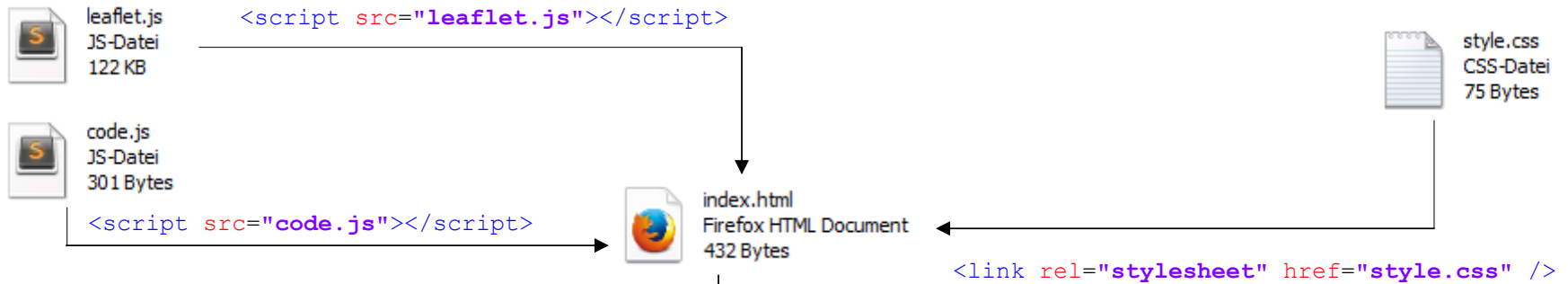
Events

Event	Data	Description
click	MouseEvent	Fired when the user clicks (or taps) the map.

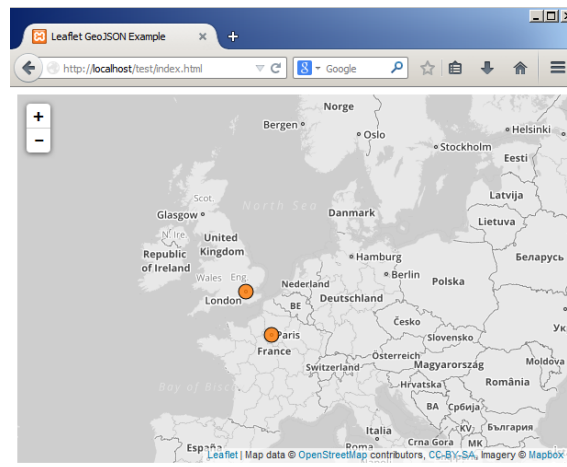
Methods

Method	Returns	Description
<code>setView(<LatLng> center, <Number> zoom?, <zoom/pan_options> options?)</code>	this	Sets the view of the map (geographical center and zoom) with the given animation options.

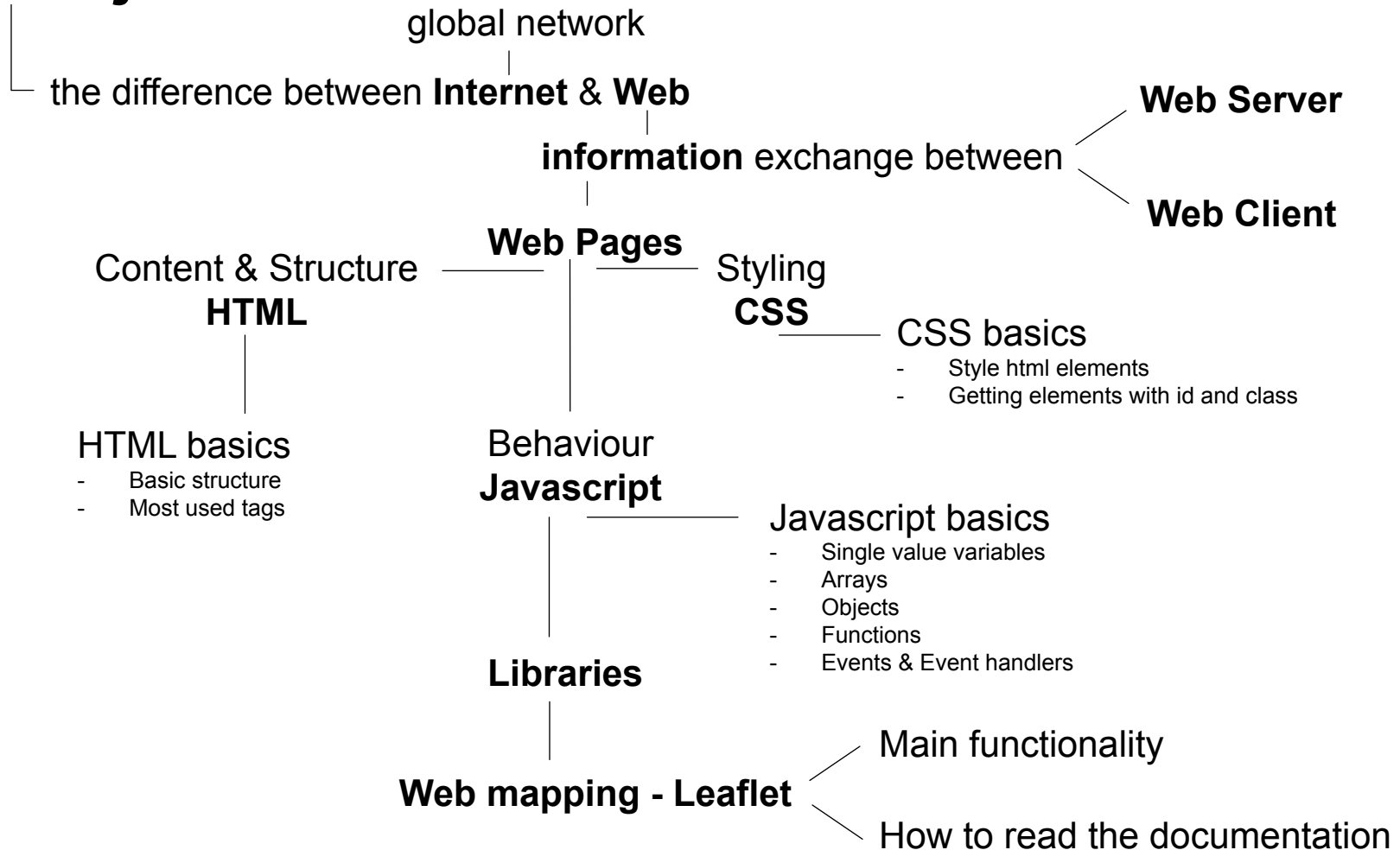
Putting it all together



.html file is displayed by the browser



We just learned about...



Questions



Exercise 1

- Using a Web Mapping Framework
 - Use a local web server
 - Understand the structure of a web site
 - Customize a web map