

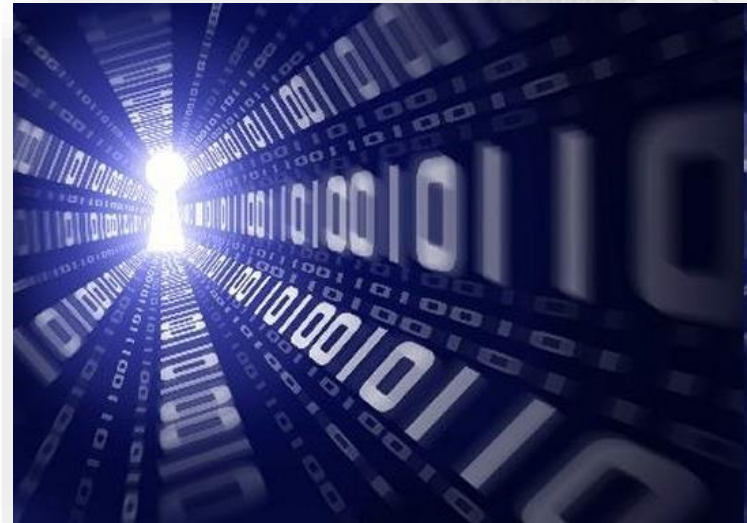
Geo-application Development

JavaScript

Client-side Scripting for Dynamic Websites

Overview

- Basic Principles
- Integration in HTML
- Functions
- DOM
- OpenLayers

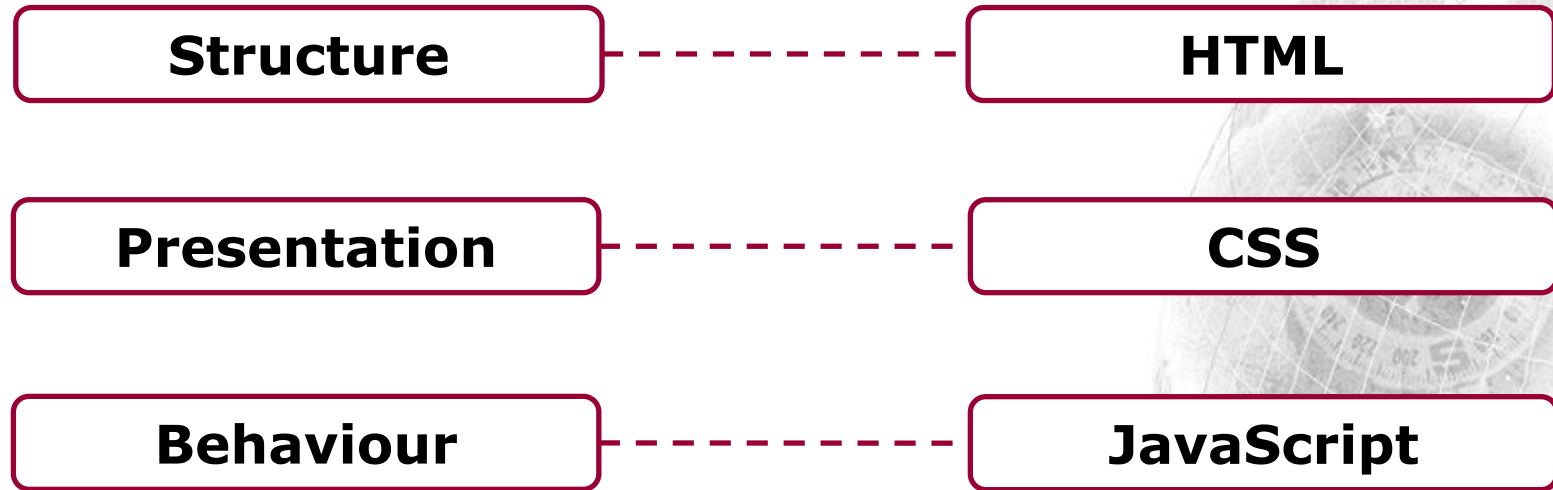


Recap of Assignment 1

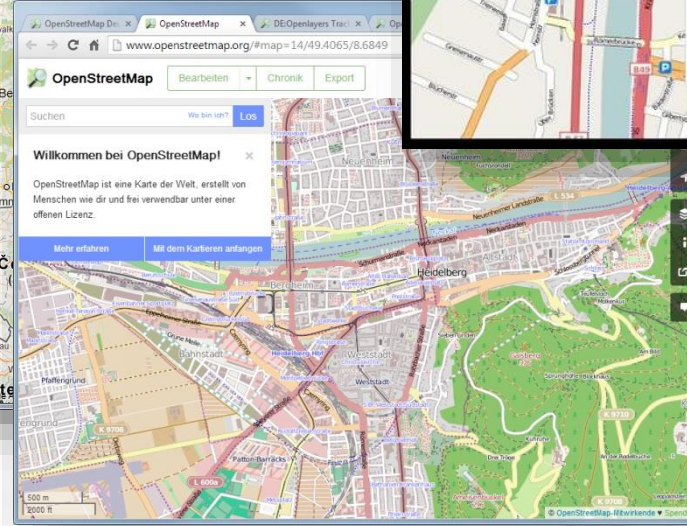
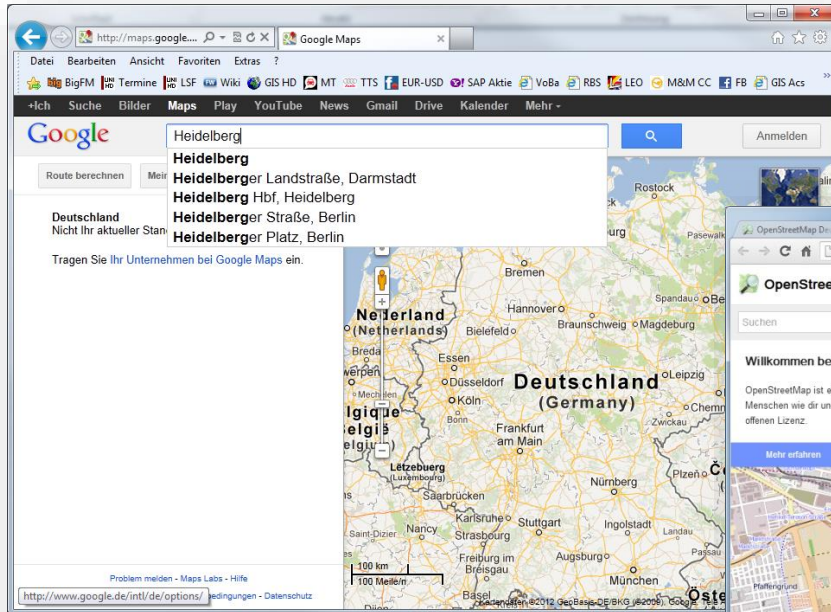
- Relative vs. absolute div height/width values
 - ◆ Or a mixture? (header/footer vs. body)
- Relative vs. absolute font size values
- Font + div interaction!



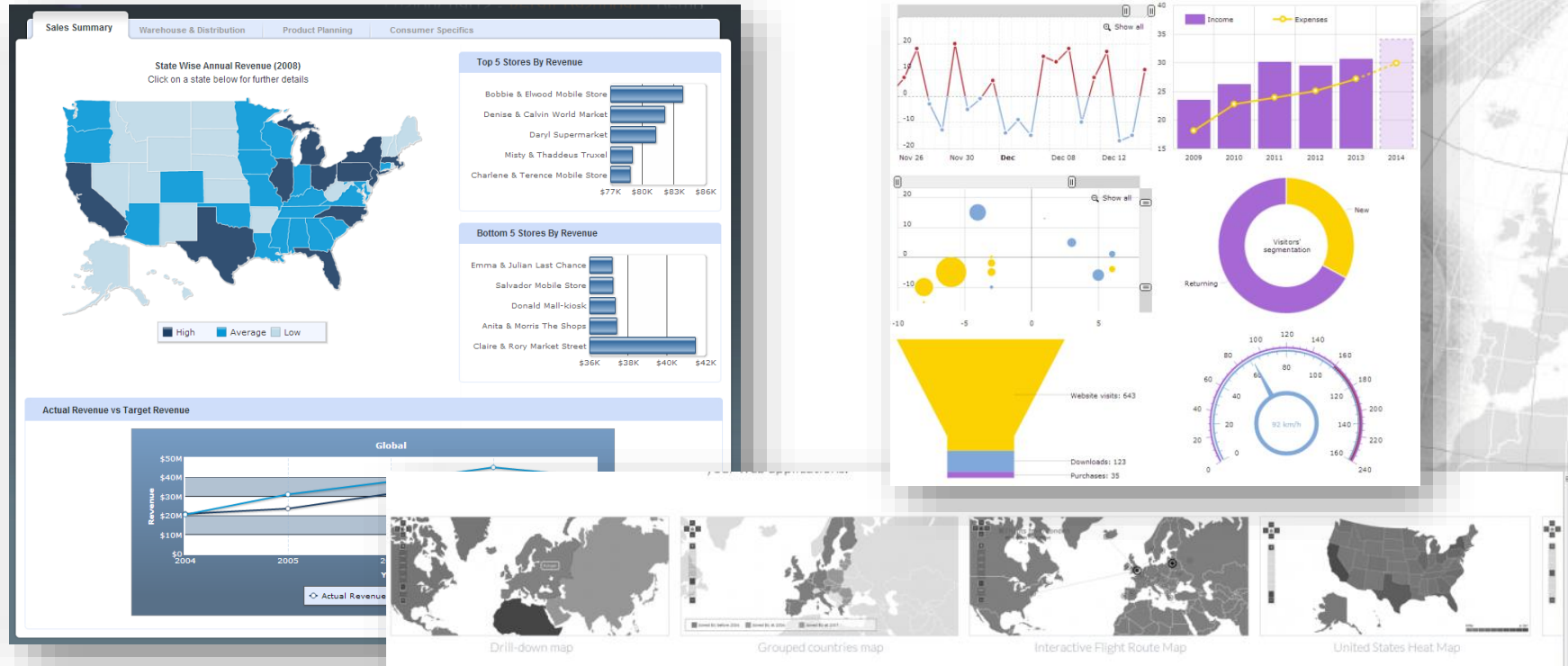
JavaScript ::: Why?



JavaScript ::: Why?



JavaScript ::: Why?



Basic Principles



JavaScript ::: Variables


Keyword `var`

name of the variable

```
var city;  
var street;  
var postcode;
```

setting the variables

```
city = „Salzburg“;  
street = „Schillerstraße“;  
postcode = 5020;
```



Variables – Data Types ::: Basics

```
var a;           // undefined
var a = 5.5;     // Number
var a = "John";  // String
```

Take care!

```
var city = "Salzburg"
var postCode = "5020";
var street = "Schillerstraße";
var houseNr = 30;
```



JavaScript ::: Text Output

```
document.write("My output");
```

```
var text = "This is my output text";  
document.write(text);
```

```
document.write("My output is: " + text);
```



JavaScript ::: Text Output

```
var text1 = "My output is: ";  
var text2 = "Hello World!";  
  
document.write(text1 + text2);
```



JavaScript ::: Conditional Statements

```
var x = 10;  
var y = 20;  
var max;
```

```
if (x > y) {  
    max = x;  
}  
else {  
    max = y;  
}
```



JavaScript ::: Comments

// a comment for one line

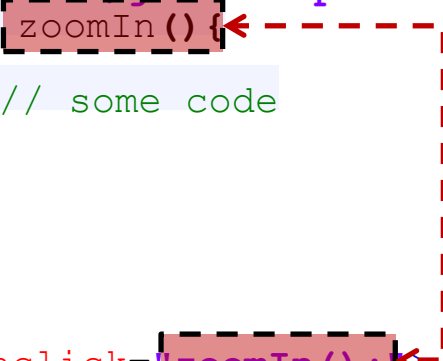
/ a comment over
several lines */*

/ a very long comment is also
possible. Documentation is always
very important – especially for
your assignments :-) */*



JavaScript ::: Functions and Events

```
<head>
...
<script type="text/javascript">
  function zoomIn() {
    // some code
  }
</script>
...
</head>
<body>
...
<a href="#" onclick="zoomIn();" >
  
</a>
...
</body>
```

A dashed red line with arrowheads at both ends connects the `zoomIn()` call in the `onclick` attribute of the `<a>` tag to the `zoomIn()` definition inside the `<script>` block.

Integration in HTML



Integration ::: In HTML

```
<script type="text/javascript">  
  var a = 10;  
  var b = 7;  
  
  var c = a + b;  
  
  document.write(c); // Comment: c is 17  
</script>
```



Integration ::: External File (*.js)

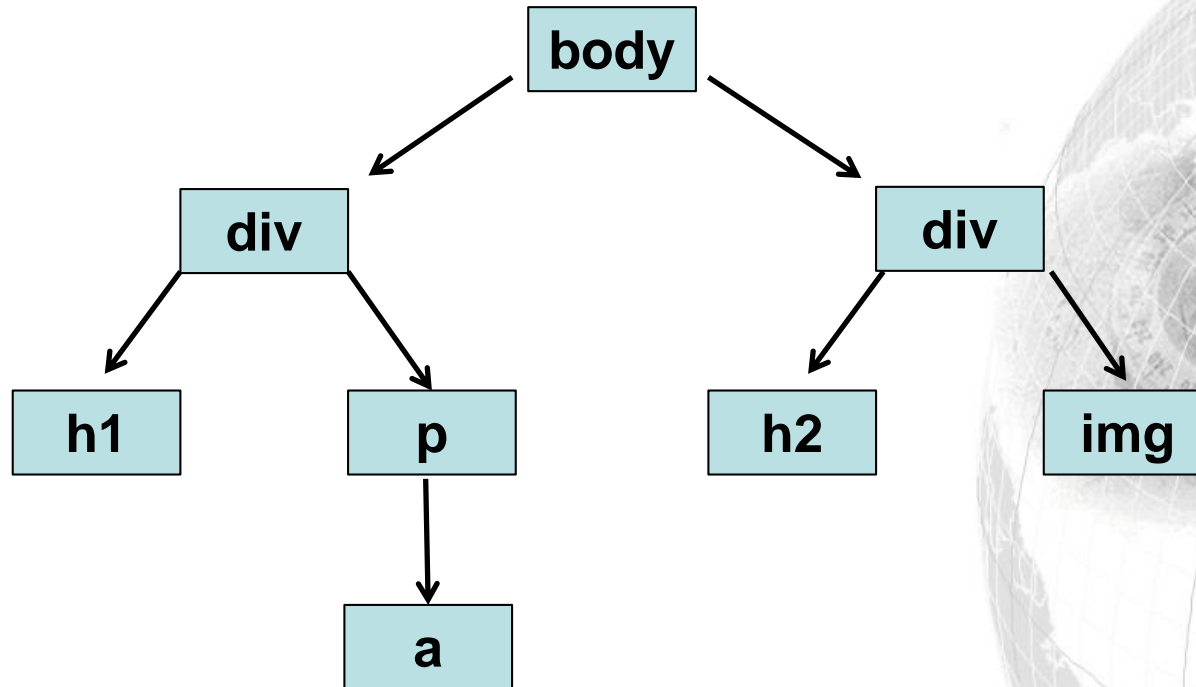
```
<script type="text/javascript" src="script.js"></script>
```

- Functionality in the external file can be used as if it were defined in the same file

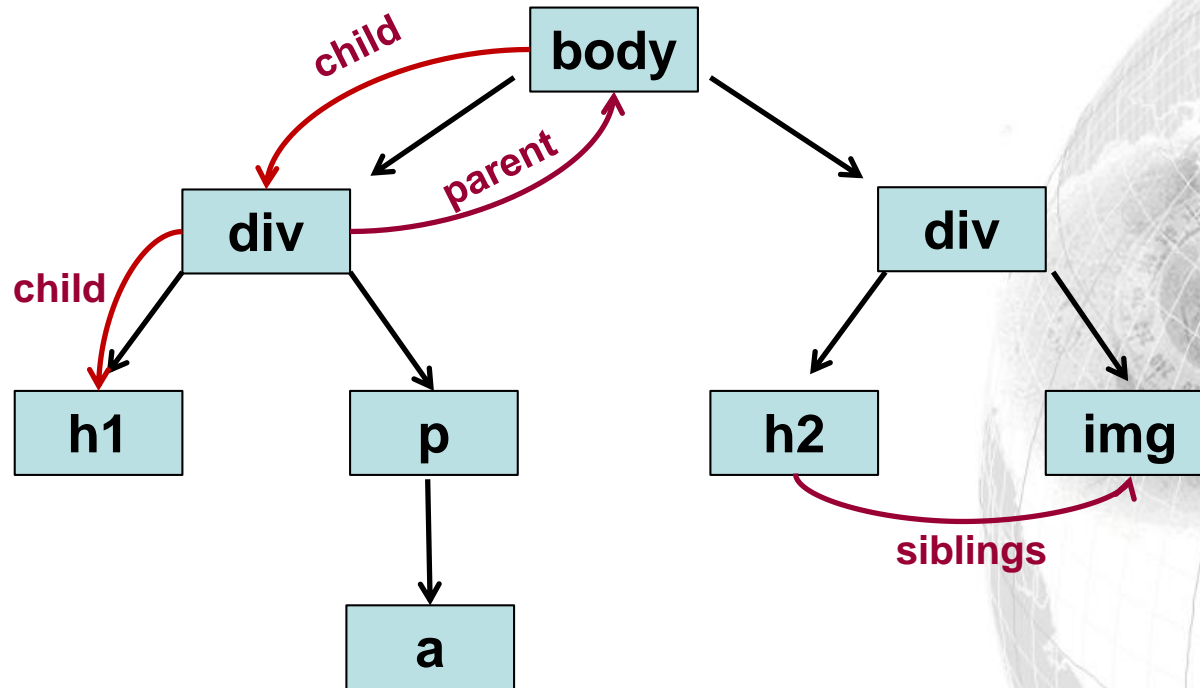
JavaScript ::: DOM Example

```
<body>
  <div id="intro">
    <h1>An OpenStreetMap Map</h1>
    <p>OpenStreetMap is a Web 2.0 project, aiming at collecting
      free geodata (e.g., Open Data).
      (Source: <a href="http://de.wikipedia.org/wiki/
        OpenStreetMap">Wikipedia</a>).</p>
  </div>
  <div id="map">
    <h2>Map</h2>
    
  </div>
</body>
```

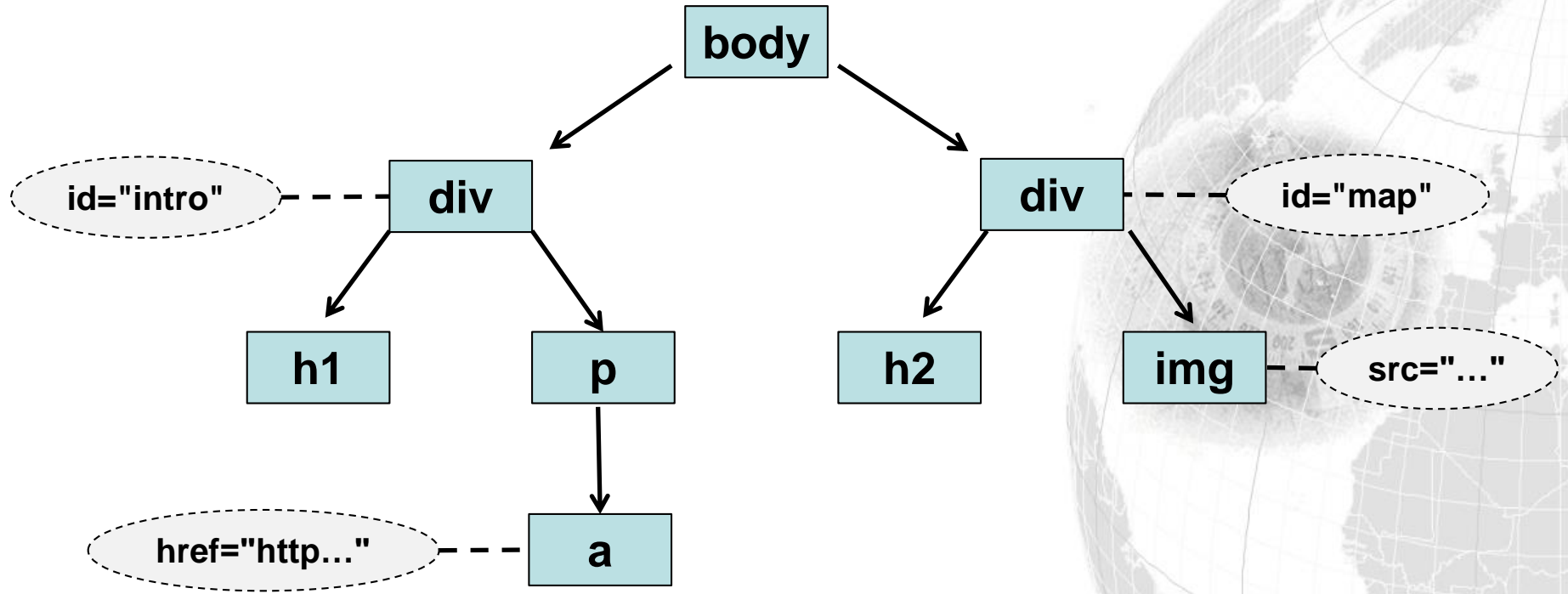
DOM ::: Nodes - Hierarchy



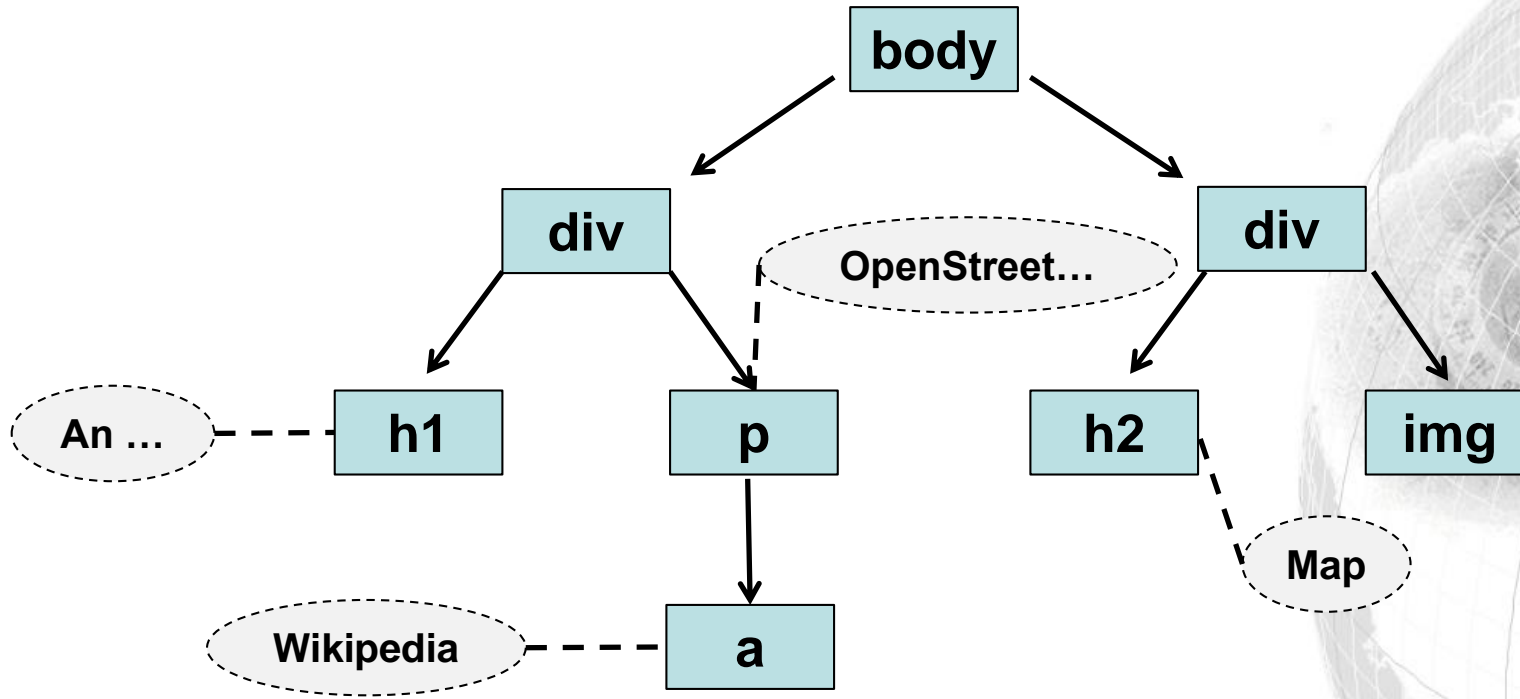
DOM – Nodes :: Childs, Parents and Siblings



DOM – Nodes ::: Attributes



DOM – Nodes ::: Text



JavaScript ::: DOM Example

```
<body>
  <div id="intro">
    <h1>An OpenStreetMap Map</h1>
    <p>OpenStreetMap is a Web 2.0 project, aiming at collecting
      free geodata (e.g., Open Data).
      (Source: <a href="http://de.wikipedia.org/wiki/
        OpenStreetMap">Wikipedia</a>).</p>
  </div>
  <div id="map">
    <h2>Map</h2>
    
  </div>
</body>
```

DOM ::: Get Elements and Attributes

```
var imgElement = document.getElementById("mapImage");
```

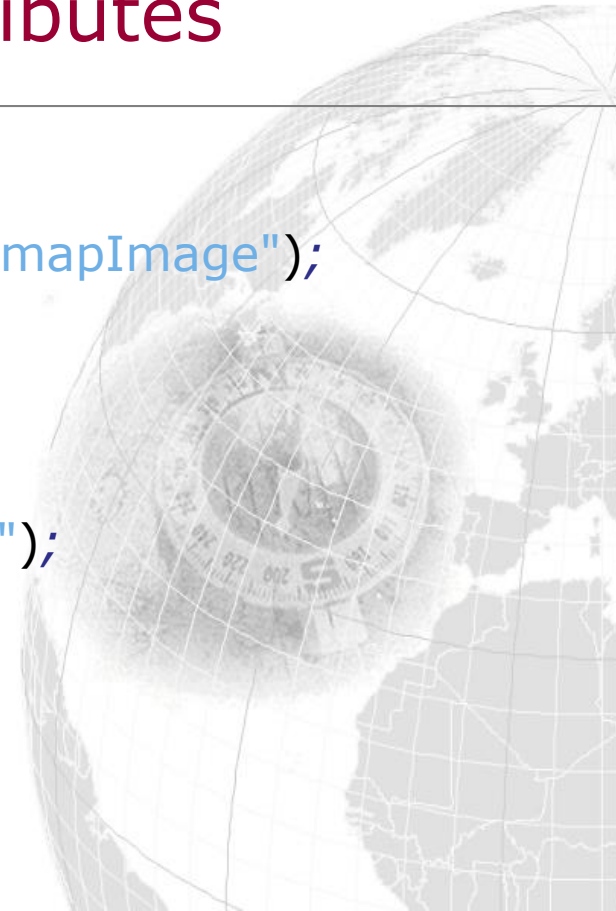
```
var imgSrc = imgElement.getAttribute("src");
```

```
// imgSrc now contains „map.png“
```

```
var imgClass = imgElement.getAttribute("class");
```

```
// imgClass now contains maps
```

```
→ if (imgClass == "maps") { ... }
```



DOM ::: Set Attributes of an Element

```
var imgElement =  
    document.getElementById("mapImage");  
  
// use another map image  
imgElement.setAttribute("src", "map2.png");  
imgElement.setAttribute("class", "myredmap");
```



DOM ::: Set Attributes of an Element

```
var divElement = document.getElementById("intro");  
divElement.style.visibility = "hidden";  
// changes visibility of intro (div) to hidden
```

```
var divElement2 = document.getElementById("map");  
divElement2.style.visibility = "visible";  
// changes visibility of map to visible
```



Useful Links

- <http://www.w3.org/DOM/>
- <http://de.selfhtml.org/dhtml/modelle/dom.htm>
- [**http://www.w3schools.com/js/js_htmlDOM.asp**](http://www.w3schools.com/js/js_htmlDOM.asp)
- Online JavaScript Tester
<http://www.webtoolkitonline.com/javascript-tester.html>

Home Study

- www.tutorialspoint.com/javascript
 - ◆ Go through the entire tutorial (JavaScript Basics)

Javascript Basics Tutorial

- JavaScript - Home
- JavaScript - Overview
- JavaScript - Syntax
- JavaScript - Enabling
- JavaScript - Placement
- JavaScript - Variables
- JavaScript - Operators
- JavaScript - If...Else
- JavaScript - Switch Case
- JavaScript - While Loop
- JavaScript - For Loop
- JavaScript - For...in
- JavaScript - Loop Control
- JavaScript - Functions
- JavaScript - Events
- JavaScript - Cookies
- JavaScript - Page Redirect
- JavaScript - Dialog Boxes
- JavaScript - Void Keyword
- JavaScript - Page Printing

Previous Page Next Page

Javascript Tutorial

[PDF Version](#) [Quick Guide](#) [Resources](#) [Job Search](#) [Discussion](#)

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Audience

This tutorial has been prepared for JavaScript beginners to help them understand the basic functionality of JavaScript to build dynamic web pages and web applications.

Prerequisites

For this tutorial, it is assumed that the reader have a prior knowledge of HTML coding. It would help if the reader had some prior exposure to object-oriented programming concepts and a general idea on creating online applications.

Execute JavaScript Online

For most of the examples given in this tutorial you will find Try it option, so just make use of this option to execute your JavaScript programs at the spot and enjoy your learning.

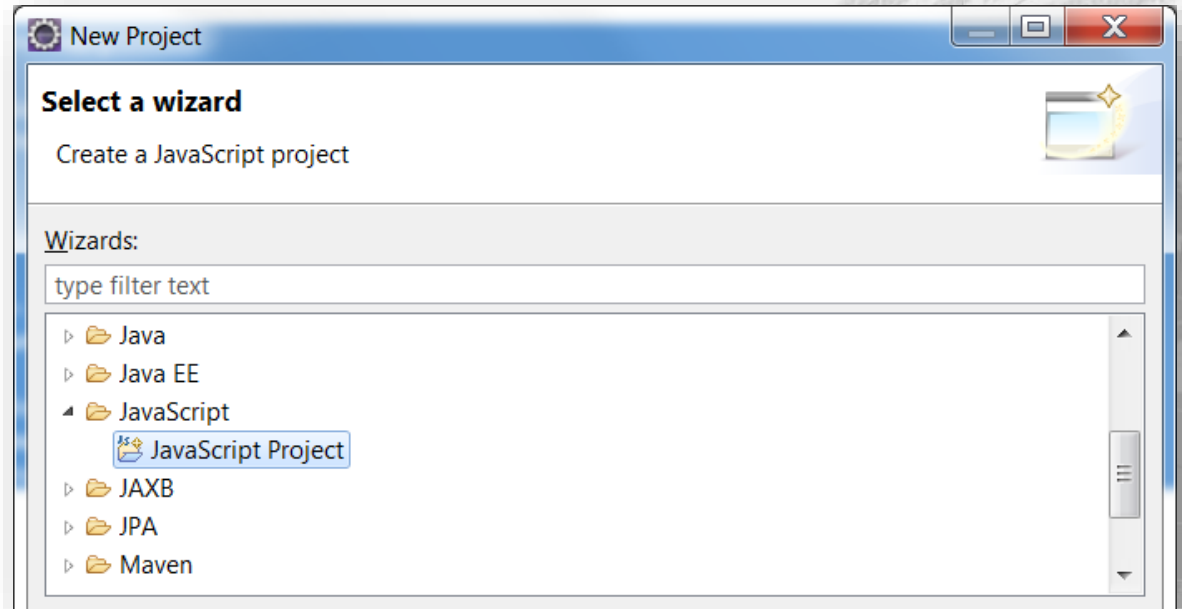
Try following example using Try it option available at the top right corner of the below sample code box –

Exercise: Mobile Web Map



Exercise: Mobile Web Map

- Create a new JavaScript project
 - ◆ “unit3”



-



Exercise: Mobile Web Map

- Use the skeleton file you developed earlier:
 - ◆ Make a copy
 - ◆ Rename the copy to *map_<name>.html*



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:

<http://openlayers.org/en/latest/apidoc>

- ♦ Hint: do not edit many parts in the code at once
 - Add a small part and test it
 - ➔ Otherwise, debugging will be difficult



Exercise: Mobile Web Map

- Use this quickstart guide:
<https://openlayers.org/en/latest/doc/quickstart.html>
 - Code to be found here
<https://openlayers.org/download>
- ➔ Please use the latest API version
→ *(no modules!) and JS tags*



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ♦ Put your JS code in a function that is called through the body's "onload" attribute
 - ♦ Background map (*e.g., OSM – as an OpenLayers layer, not WMS*)
 - ♦ Centre the map at WGS84 position 13.06072 E, 47.78869 N
 - The spatial reference system "EPSG:3857" is used for the map (!)
→ reprojection necessary

Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ◆ Add map controls (navigation buttons, overview map, scale bar, etc.)
 - ➔ Approach:
 - ➔ Create a scale line control object (`ol.control.ScaleLine`)
 - ➔ Add it to the map → extend default controls



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ♦ Add a marker to the centre position
 - OL example “Icon Symbolizer”
- ➔ Approach:
 - ➔ Objects: Feature → Style → VectorSource → VectorLayer
 - ➔ Set style (assign style to feature)
 - ➔ Add the layer to the map



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ◆ Create a function “drawMarker(posMarker)” that...
 - ...takes a position (Coordinate object) as an input parameter
 - ...draws a marker at this position
(adds a Feature to a Vector source)
 - ➔ Call the function with the position created earlier

Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ◆ Draw a marker on your computer's current location using the HTML5 geolocation functionality
 - Add a marker to the current position
 - Use the function "drawMarker()" to draw the marker



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ◆ Draw a marker on your computer's current location using the HTML5 geolocation functionality
- ➔ Approach:
 - ➔ Create ol.Geolocation object, call geolocation.on() function
 - ➔ Call draw marker function



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ♦ Add Bing Maps layers
 - Layers: Road, Aerial and AerialWithLabels
 - Key required:
 - ➔ s. Blackboard under Content → Part 1 → Exercises



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ◆ Add Bing Maps layers
 - ➔ Approach:
 - ➔ Create layers array
 - ➔ Add ("push") ol.layer.Tile objects (incl. source and key) to the array
 - ➔ Add layers array to the map

Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ♦ Add a layer switcher
 - <https://github.com/walkermatt/ol-layerswitcher>
 - ➔ Download and integrate local JS and CSS files
 - ➔ !!! Download the JS file from the “**dist**” folder !!!



Exercise: Mobile Web Map

- Extend your file *map_<name>.html* by a simple OpenLayers map:
 - ◆ Add a layer switcher
 - ➔ Approach:
 - ➔ Download and integrate JS and CSS file into your HTML file
 - ➔ Create two layer groups (`ol.layer.Group`) for your `ol.Map` object
 - ➔ Create a `ol.control.LayerSwitcher` object and add it to your map

Exercise: Mobile Web Map

- Add a WMS layer to the map
 - ◆ WMS: <http://data.stadt-salzburg.at/geodaten/wms>
 - ◆ Layer “forschung”
 - ◆ Transparent background



Exercise: Mobile Web Map

- Add a WFS layer to the map
 - ◆ WFS:
<https://dservices.arcgis.com/Sf0q24s0oDKgX14j/arcgis/services/ParkingSpaces/WFSServer>
 - Layer "ParkingSpaces:ParkingSpaces"
 - Set the "maxFeatures" parameter to 20
 - Version 2.0.0



Exercise: Mobile Web Map

- Add a WFS layer to the map
 - ◆ WFS: <https://data.wien.gv.at/daten/geo>
 - Layer “ogdwien:KURZPARKZONEOGD”



Exercise: Mobile Web Map

- Add the contents of a local Flickr file to the map
 - ◆ File download link:
http://berndresch.com/download/work/flickr_salzburg.json
 - Display the Flickr posts on the map
 - Extract *date*, *text* and *photo_url* parameters
 - Take this approach for parsing the JSON data:
 - 1.) *JSON.parse(flickrJSONString)*
 - 2.) iterate over the resulting array and create an array of *ol.Feature* objects (*then add the attribute values*)

Exercise: Mobile Web Map

- Add the contents of a local Flickr file to the map
 - ◆ File download link:
http://berndresch.com/download/work/flickr_salzburg.json
 - Use the Flickr logo as the point icon
 - If a Flickr icon is clicked, display its metadata in a FramedCloud
 - Integrate *date*, *text* and the image into the cloud

Exercise: Mobile Web Map

- Note: you are welcome to design the topic for your end-of-term assignment at any time
 - JavaScript vs. Python?



Exercise: Mobile Web Map

- Modify your function *drawMarker(posMarker)* to *drawMarker(posMarker, height, width, icon)*, where...
 - ◆ ...*posMarker* is an OpenLayers.LonLat object
 - ◆ ...*height* and *width* are integer values
 - ◆ ...*icon* is a String denoting the icon to be displayed
- ➔ Adapt the function calls accordingly!



Exercise: Mobile Web Map

- Assignment 2



Geo-application Development

JavaScript

Client-side Scripting for Dynamic Websites