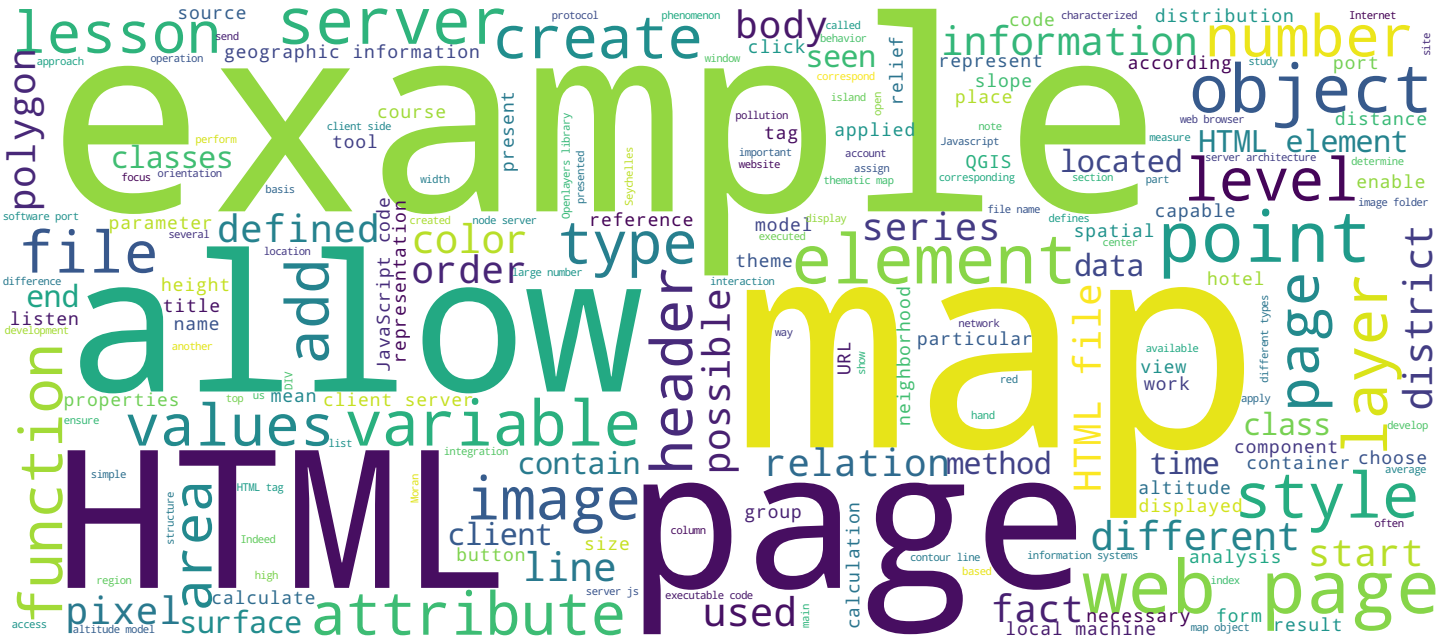
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

RESCIF
Réseau d'excellence des sciences de l'ingénieur de la Francophonie

An Introduction to Geographic Information Systems

Dynamic and Interactive Web-based Cartography

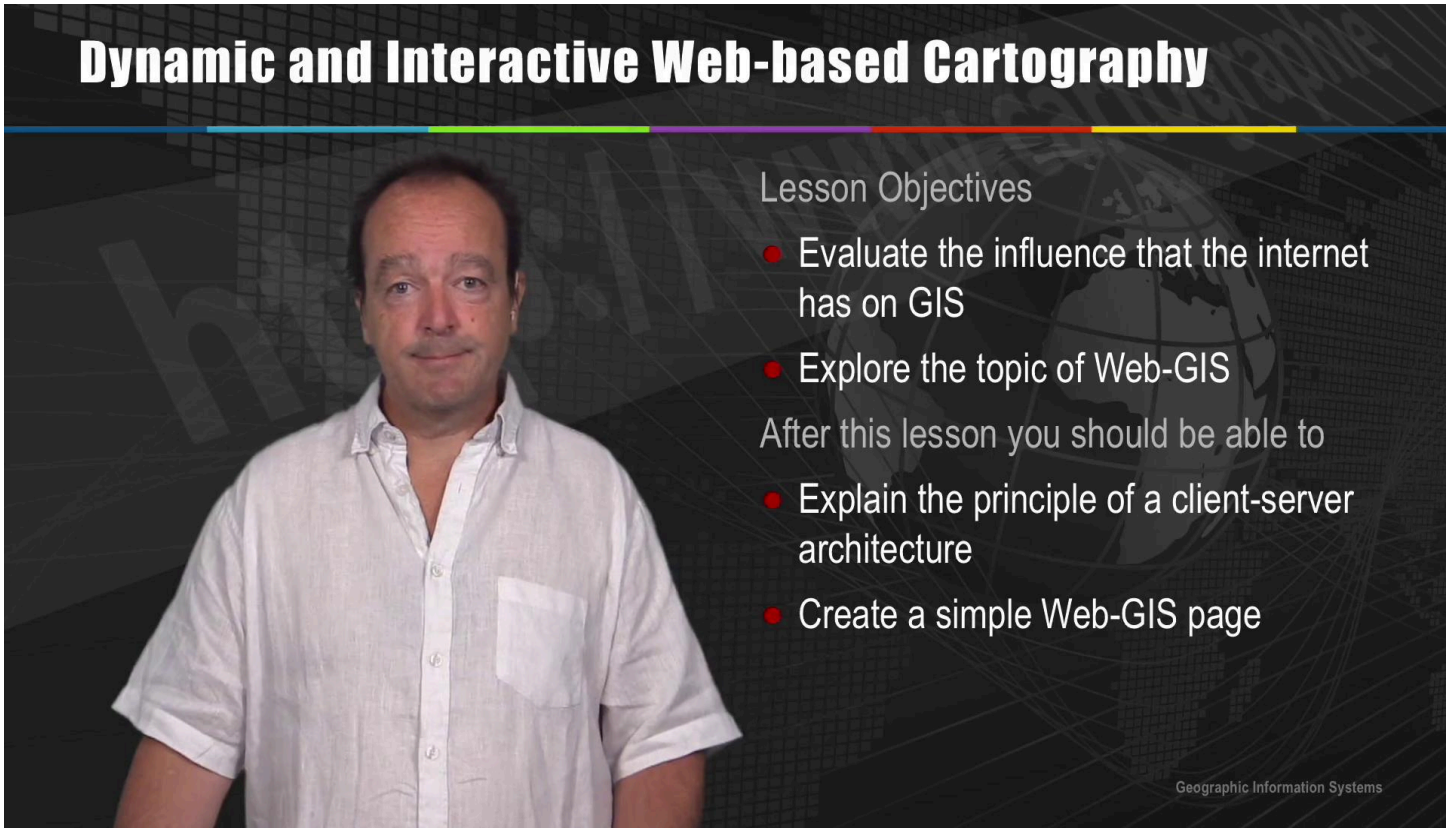
Stéphane Joost, Marc Soutter, Fernand Kouamé, Amadou Sall



Search MOOC

Video

EPFL

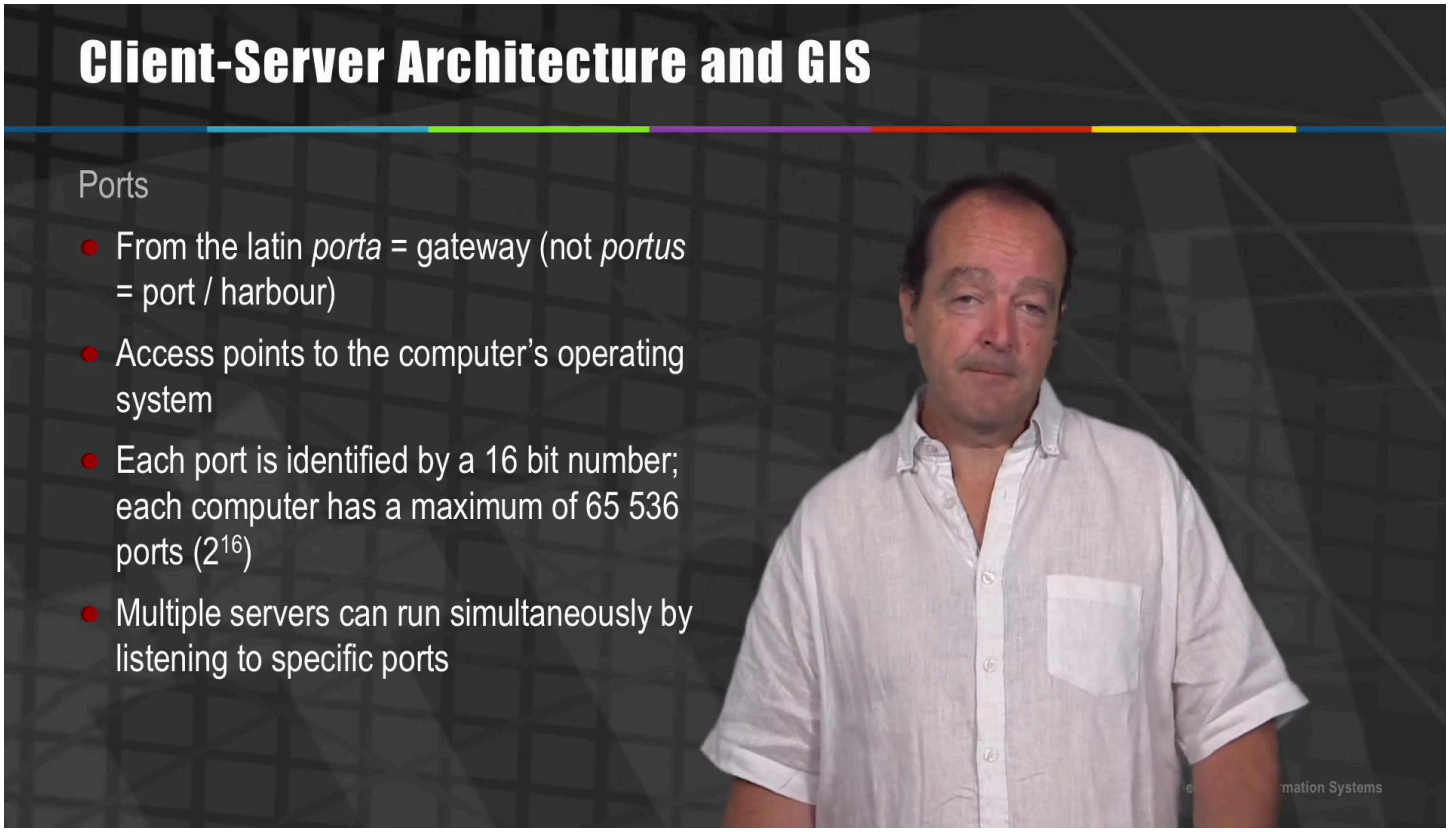


So here I am again and it is with great pleasure that I see you again for this lesson which will focus on certain aspects of the relation between mapping and the Internet. This is a topical issue since more and more often, computer applications, and in particular applications in the world of geographic information systems, rely on the Internet to offer multi-platform solutions, usable as much on desktop computers than on tablets or smartphones and which allow to use different types of operating system, whether it is in Windows, MacOS, Linux, Android, and so on. The objective of this lesson is to discuss generally, and inevitably a little superficially, the theme of relations between geographic information systems and the Internet, then talk in more details about the theme of integration of dynamic and interactive mapping elements in web pages, which is generally called GIS web. At the end of the lesson, you should be able to describe the principles of client-server architectures, and you should be able to create a simple web page containing a map element.

Notes

Summary

0m 18s

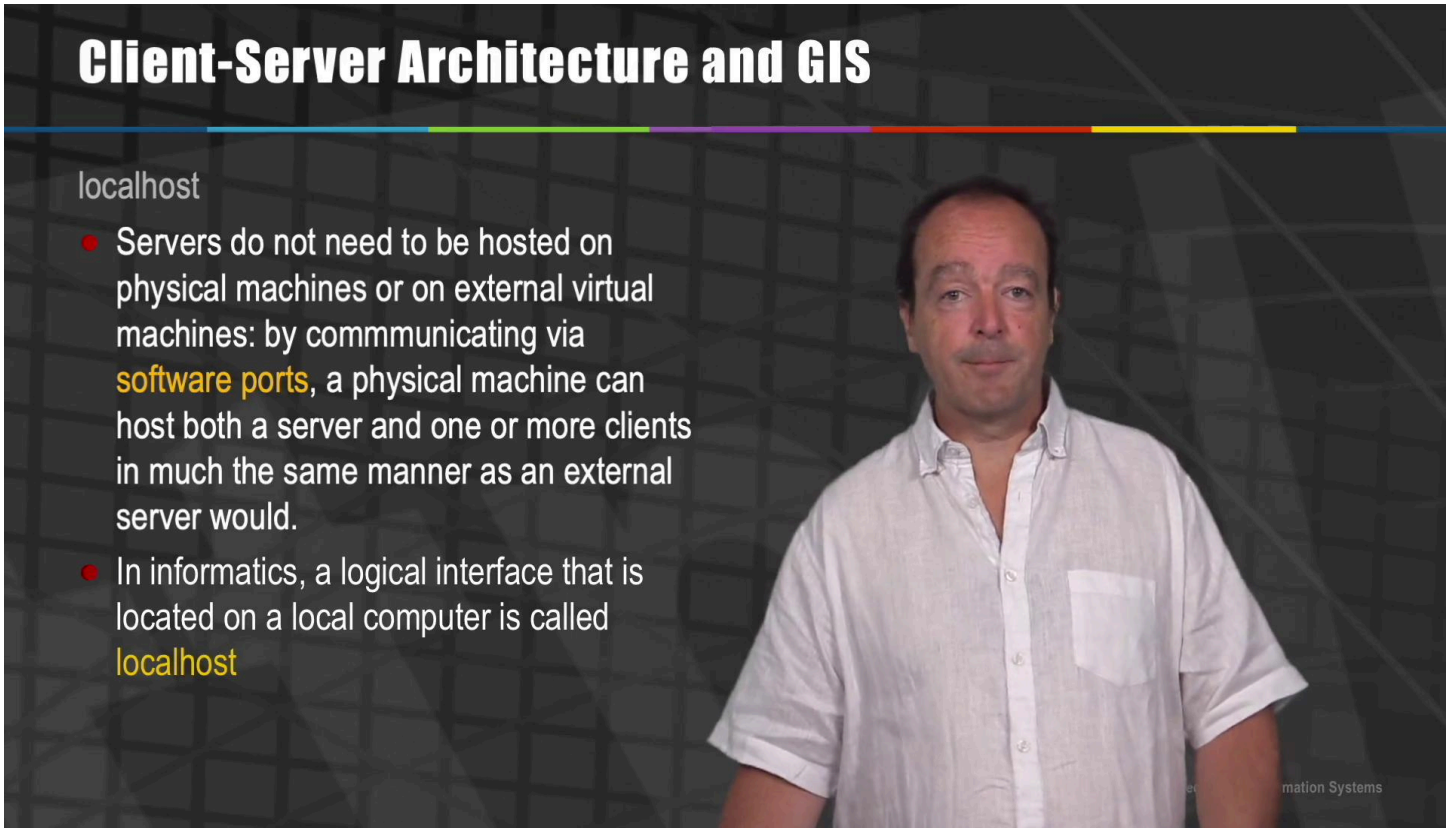


We will therefore discuss successively the theme of client-server architecture in relation to geographic information systems, then we will focus on the fundamental elements of the web page, that are the HTML and CSS language, which are the beginnings which will allow us to then adress the topic of cartography in web pages, so the theme of the GIS web, and we will end the lesson with an example of application, of realization of a web page which contains a dynamic and interactive cartographic element. A server is a software or a machine which is capable of listening to the network and which is capable of receiving queries from different clients and to respond to these queries. So the goal of a server is to allow the data sharing or the material or software resources sharing between several clients in a architecture qualified as client-server architecture. There are a very large number of different server types: database servers, file servers, mail servers, web servers, game servers, and so on. The servers therefore listen to the traffic on the Internet in order to identify requests addressed to them, and the portal that allows them to listen on the network is what is called a software port, which is actually the access point to the operating system of a computer.

Notes

Summary

1m 43s



Software ports are numbered from 1 to 65,536 and on the same machine, in fact, we can have several server softwares which operate simultaneously, with each listening or working with one or two specific software ports, often a port for the incoming traffic and a second port for the outgoing traffic. As an example of frequently used port numbers, we can mention ports 20/21 which are used for protocols to transfer FTP file 20 for the incoming flow 21 for the outgoing flow, port 80 to browse web pages on an HTTP web server, port 443 in case this web server is secured with the HTTPS protocol, or port 5'432 for the connection to a PostgreSQL database. The server softwares do not have to be hosted on a remote machine but can be hosted on the local machine at the same time as client softwares, which is particularly useful if we do development. It is simply necessary that the operating conditions are the same as if the servers were on a remote machine, namely that the communication between the client and the server goes through the same software ports as if these servers were on a remote machine. The local machine, in a URL address, is designated by the "localhost" term which in fact designates the logical interface of this local machine.

Notes

Summary

3m 24s

Client-Server Architecture and GIS

- The name localhost is associated with IPv6 ::1 and the IPv4 127.0.0.0/8 address range (all addresses located between 127.0.0.1 and 127.255.255.255)
- For example: http://localhost = http://127.0.0.1
- The address for port 3000 on the local machine: http://localhost:3000

Geographic Information Systems

The localhost address is also associated to a series of IP addresses. In the case of IP version 4, it is the address 127.0.0.1 that is the most frequently used, but the series of the following addresses is also usable. Finally, in a URL address, if we want to specify a particular port, we simply add two points and the port number to target that port, so if we want to tap on port 3000 of a local machine, the URL would be http://localhost:3000 A client-server architecture is therefore generally constituted by a server software that listens to traffic on the network, which is capable of identifying a request addressed to it and then send back a reply.

Notes

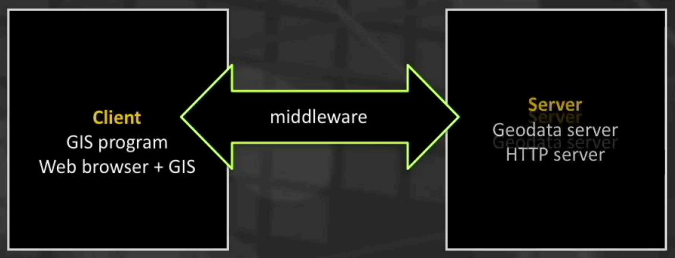
Summary

5m 10s

Client-Server Architecture and GIS

In the GIS context

- Server
  - Geodata server(s)
  - HTTP server
- Client
  - GIS program
  - Web browser + GIS
- Client-server communication
  - Middleware



Geographic Information Systems

On the other side, we have a client software which is capable of formulating requests, to send them to the server and then to retrieve the response from the server and to interpret it to display the information that was transmitted. Finally, the exchanges between client and server are often managed by an intermediate software called middleware. In the particular context of geographic information systems, the server side is often simply composed of a data server. It is quite frequent however to have two servers in series, the first one being destined to manage queries and create responses, whereas the second one is limited to the provision of data. This structure has advantages from the security point of view since the access to data is only allowed for the intermediate server, which can manage the access rights of the different clients. On the client side, we will have a user interface composed either of a GIS software, QGIS, Manifold, Mapinfo, ArcGIS, etc. or a web browser including a map component, an association which is generally called GIS web. And again, we will find various communication protocols between client and server.

Notes

Summary

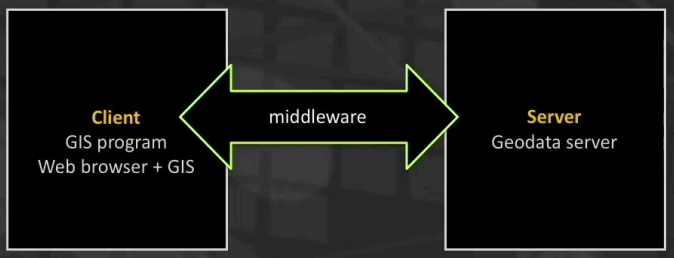
6m 09s

Client-Server Architecture and GIS

Examples of middleware

ODBC – Open Data Base Connectivity

- Transfers SQL queries generated by the client to the server
- Generic process that is valid for all ODBC databases – often with a specific **ODBC** driver



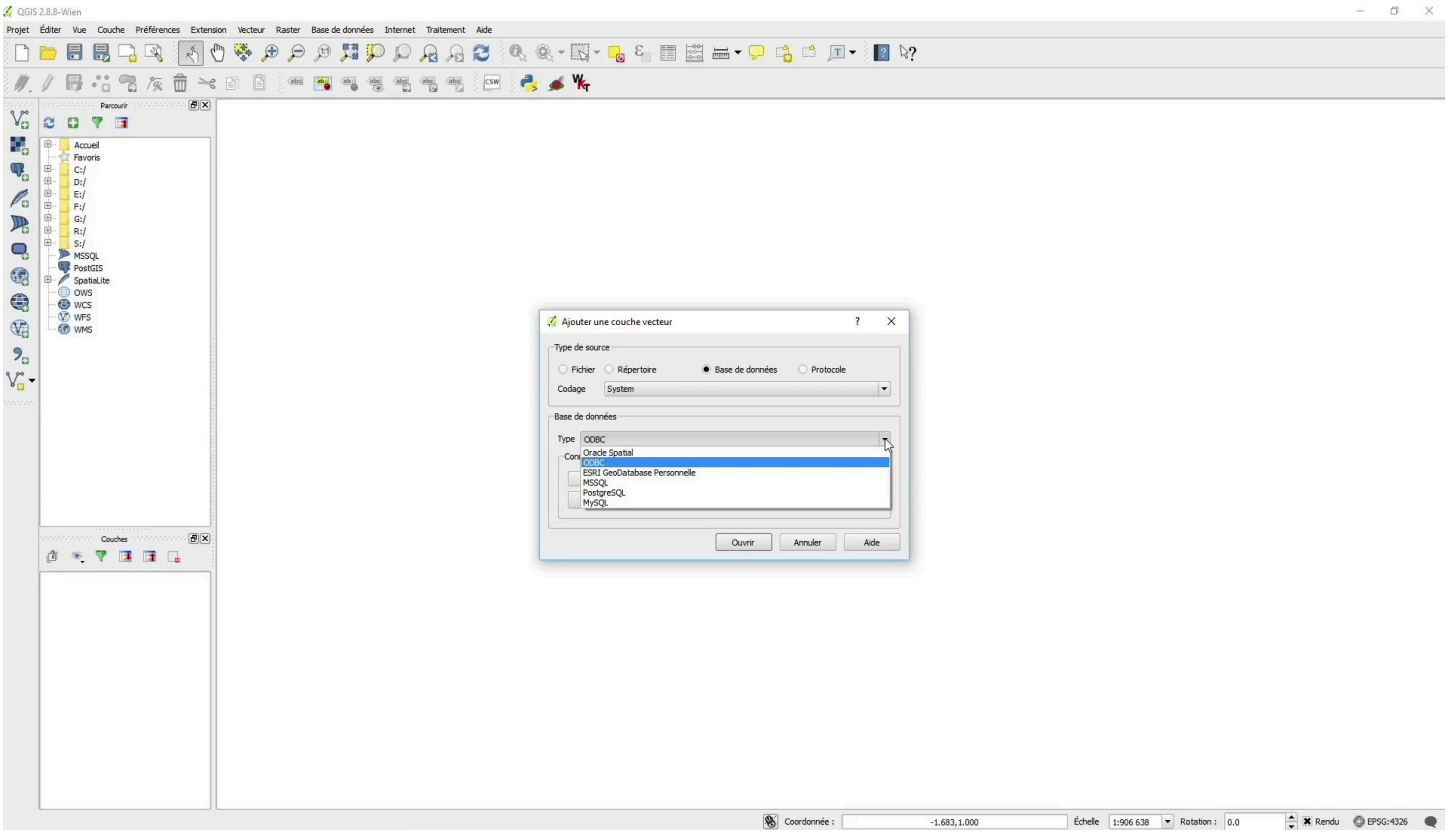
Geographic Information Systems

The most common of these softwares which manage communications between client-server is the ODBC software, for Open Data Base Connectivity. It is a software initially developed by Microsoft for Windows, and subsequently taken over by other publishers for Unix / Linux and Java platforms. Data manipulation requests are formulated in SQL language in the client software, and then transmitted to the data server through ODBC. The scheme of operation implies that manufacturers of database management software had to develop specific ODBC drivers for each database management system.

Notes

Summary

7m 28s

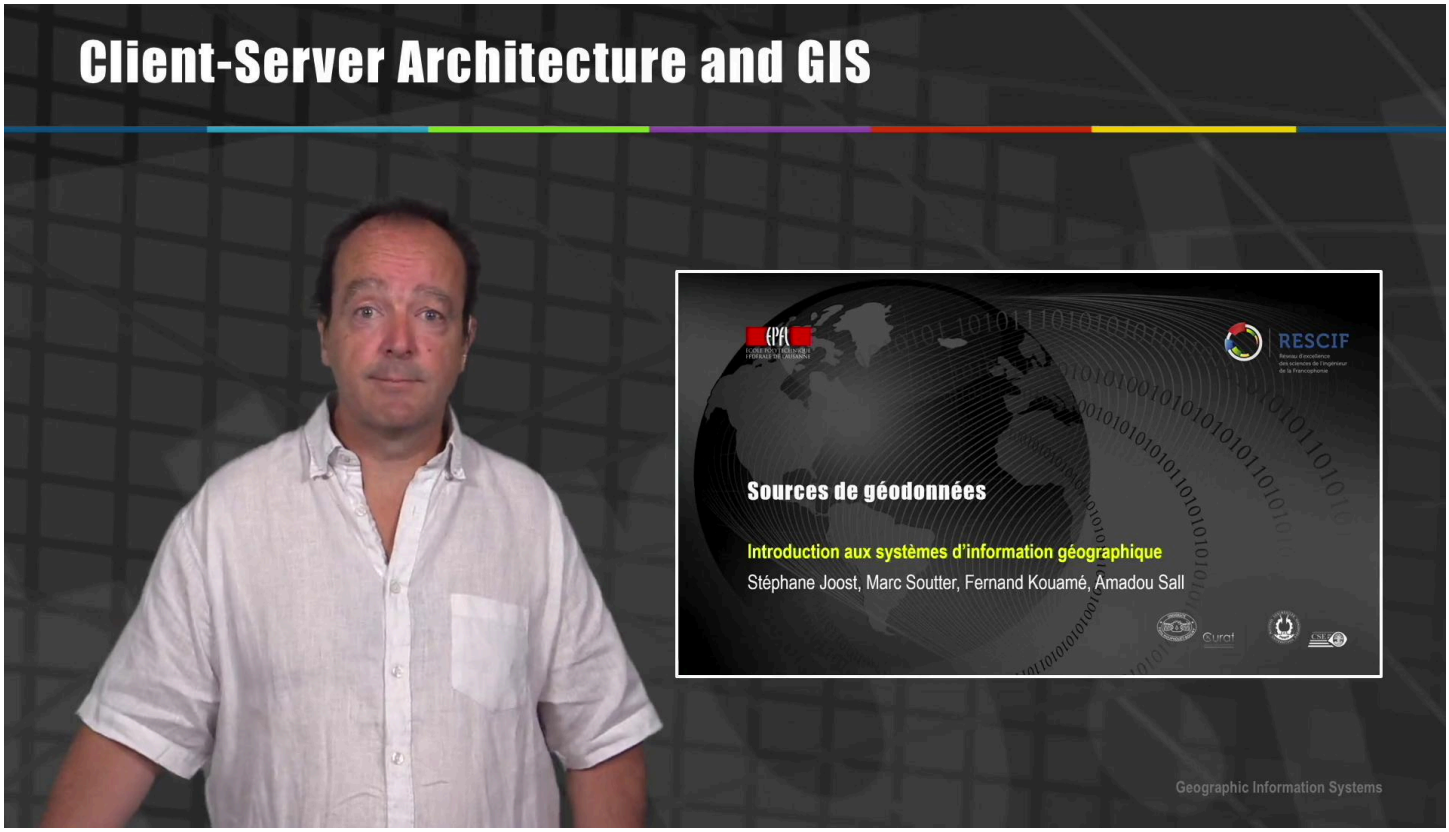


We see here in the case of QGIS that when a vectorial layer is added for example, we have the possibility of searching this layer in a database, and the types of database connections that are offered include different databases from different manufacturers, so Oracle, MSSQL, Microsoft, Postgres, MySQL, but also the generic ODBC driver which allows to connect to databases using this particular protocol.

Notes

Summary

8m 10s



Two other examples of middleware quite frequently encountered in Windows: OLEDB, for Object Linking and Embedding Database, which is in fact a successor of ODBC based on the COM interfaces, and ADO.NET, which is the data access component of the Microsoft.NET framework. Finally, in the course devoted to geodata sources, we had already talked a bit about the HTTP protocol, in particular to interrogate geoservices, so WMS and WFS servers.

Notes

Summary

8m 46s

Webpage: HTML and CSS

HTML – HyperText Markup Language

- A webpage is an HTML document: a simple annotated text file that defines different types of textual elements (headers, paragraphs, titles, links, images, etc.)
- The annotations (markup) are identified by tags, i.e. key-words placed between angled brackets that indicate the start < > and end </ > of an HTML element

```
<html>
    <head>
        <title> e-Atlas de l'Orontes </title>
    </head>
    <body>
        <p> Bienvenue sur l'e-Atlas de l'Orontes </p>
        <img  src="sources.jpg"
              alt="Photo des sources"
              width="220"  />
    </body>
</html>
```

Geographic Information Systems

In the particular case of the HTTP protocol, we have, on the server side, either an HTTP server, or geoservices such as Web Map Service or Web Feature Service. On the client side, a web browser with a GIS component or, more occasionally, the GIS software, which will not usually use an HTTP server but which can look for tiles in geoservices as seen in the course on geodata sources. The communication in the client-server sense relies on the HTTP protocol, which can possibly include parameters, as we had seen to define the elements the server must return and in the client sense, we will simply find a flow of information in the form of HTML, images, JSON and GML files, etc. In the rest of the course, we will now focus on the case where the client is composed of a web browser which includes geographic information system functionalities. Before addressing the question of the integration of geographic components into a web page, it is necessary to remind us a little bit of the foundations of the organization and construction of a web page, so the HTML language for HyperText Markup Language and the CSS for style elements. This is the object of this second part of the course, an accelerated review of the basic principles of this structuring of web pages.

Notes

Summary

9m 20s

Webpage: HTML and CSS

HTML documents must:

- Begin and finish with the tags <html></html>
- Contain a header section <head></head> and a content section <body></body>
- HTML is not case sensitive, but lowercase is standard

```
<html>
    <head>
        <title> e-Atlas de l'Orontes </title>
    </head>
    <body>
        <p> Bienvenue sur l'e-Atlas de l'Orontes </p>
        <img  src="sources.jpg"
              alt="Photo des sources"
              width="220"  />
    </body>
</html>
```

Geographic Information Systems

A web page is primarily an HTML document, so an.HTML file, which is an annotated text file in order to define different types of content, the header, footers, paragraphs, titles, etc. These annotations, which are called Markup are formed by tags which are in fact keywords placed between chevrons and which signal the start and the end of a content element or HTML element. In the example we have before us, we see that the main tag is the HTML tag that actually defines the HTML page, itself subdivided into a header, with the head tag containing a title, with the title tag, header followed by a body part that is actually the body of the page, which contains a paragraph with a text, "Welcome to the e-Atlas of Orontes" and an image characterized by its properties. The access path to the image has a substitute text in case the image is not accessible and the size in width of this image. HTML documents should start and end with start and end tags, HTML tags. They must contain a header section and a content section, with the head and body tags. And finally we note that the syntax of HTML is not case-sensitive, but in general we often use lower cases everywhere to make it easy to read.

Notes

Summary

11m 19s

Webpage: HTML and CSS

- The header is used to hold metadata, references, and often the page title <title> </title>
- The HTML elements that should be visible on the webpage are stored between the body tags

```
<html>
    <head>
        <title> e-Atlas de l'Orontes </title>
    </head>
    <body>
        <p> Bienvenue sur l'e-Atlas de l'Orontes </p>
        <img  src="sources.jpg"
              alt="Photo des sources"
              width="220"  />
    </body>
</html>
```

Geographic Information Systems

And finally, as we have seen, the header contains metadata, references to libraries, to external resources, often the title of the page, and in the body we will place the different elements of content, the various HTML elements which we want to see appearing on the page itself.

Notes

Summary

12m 59s

Webpage: HTML and CSS

Examples of common html tags

- <a> hyperlink
  - to a website
  - to another element on the page (location defined by an id)

```
<a href="http://www.epfl.ch"> Site Web de l'EPFL </a><br>
<br>

<a id="top">This is the top</a><br>
<a href="#top" > Back to top </a>
```

Site Web de l'EPFL

This is the top
Back to top

Geographic Information Systems

Some examples of common tags with title tags from level one to level six, the P tag for paragraphs, and then a series of tags which allow to reinforce text in italics or in bold, the line break, the horizontal line. The HyperText link with tag A which can point either to a website when we give href as parameter, the address, the url of a website, here the example of the polytechnic school website, or to an element of the page located higher or lower which is defined by its identifier, here we have the example of a sentence, "this is the top", which is characterized by its ID which is called top and the reference refers to this place of the page, with the hashtag character that we will see later which actually targets an object that carries a top ID.

Notes

Summary

13m 20s

Webpage: HTML and CSS

Examples of common html tags

-   ordered list (numbered)
-   unordered list (bulleted)
-   list element

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

50. Coffee
51. Tea
52. Milk

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

Geographic Information Systems

The tags used to define ordered lists with the OL tag, not ordered with the UL tag and the LI tag for the list elements.

Notes

Summary

14m 22s

Webpage: HTML and CSS

Examples of common html tags

- <img> to include an image
- No closing tag
- Required attributes:
  - src access path to the image's source, as a URL or a local path (relative)
  - alt text that should be displayed in case the source is not accessible
- The attributes have to be indicated by single or double quotations

```
<img src="sources.jpg" alt="Photo des sources" width="220" />
```

Examples of paths for a source that is located:

- In the same folder as the HTML file
  src = "sources.jpg" or src = "./sources.jpg"
- In an image folder located at the same level as the HTML file
  src = "./images/sources.jpg"
- In an image folder located at the same level as an HTML folder that contains the HTML file
  src = "../images/sources.jpg"
- Stored on a webpage
  src = "http://www.mysite.ch/OrontesImg/sources.jpg"

Geographic Information Systems

The image tag which we have already seen, with the particularity that this tag does not require an end tag and, as another particularity, that its two attributes, the source and the substitution text, are mandatory attributes. And let's also note that the value of the attributes is characterized by a text taken between inverted commas or apostrophes. The definition of access paths follows the standard rules. So for example in the case of a source which is in the same folder as the HTML file, the source will be simply composed of the file name or possibly by the syntax: (dot) which designates the folder in which we are, the file name. If we have an image folder which is placed at the same level as the HTML file on which we work, the link to the source will take the following form : ./image folder/file name. It can happen that the image folder is at the same level as the HTML file which contains the HTML file on which we work, and in this case to go up of a notch in the hierarchy, we will have a syntax like :/image folder/source file.jpeg And finally in the case where the image is hosted on a remote website, we simply have the url of this image for the source.

Notes

Summary

14m 33s

Webpage: HTML and CSS

-   and   are neutral elements that are used as containers for other HTML elements and are used to structure and format the HTML document.
- The div element is framed by line breaks (block element); the span element does not have any default formatting (line element).

```
<h1>My <span style="color:red">Important</span>Heading</h1>
```

My Important Heading

```
<p>This is some text.</p>
<div style="color:#0000FF;background:lightgrey">
    <h3>This is a heading in a div element</h3>
    <p>This is some text in a div element</p>
</div>
<p>This is some text.</p>
```

This is some text.

This is some text in a div element

This is some text in a div element

This is some text.

Geographic Information Systems

The table tag which defines a table composed of lines, with the TR tag for row, each line being composed of cells described by the TD tag. We can find a complete list of HTML tags on the w3schools site, in the HTML part of this site. To conclude, the neutral elements constituted by the DIV and SPAN tags which are in fact simply containers to accommodate other HTML elements and which are mainly used to structure and stylize an HTML document. The DIV element is framed by line breaks, so it is a block element, and each subsequent element will end up in the next line, whereas the SPAN element is an on-line element, and can be applied to a line section, with here the example, at least important, which is stylized in red using a SPAN in the title phrase and then the second example, a DIV element which is colored in gray, a background in gray and then a text in blue, a style that is applied to the content of that container, so the title of level h3 and the following paragraph.

Notes

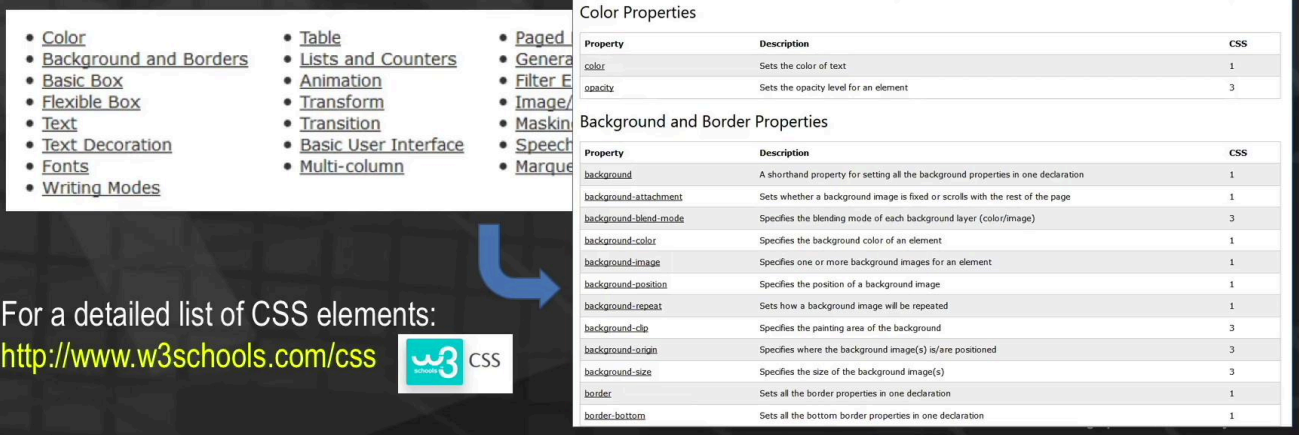
Summary

16m 12s

Webpage: HTML and CSS

HTML & CSS – Cascading Style Sheet

- All HTML elements have an associated style attribute, given by a string that is made up of a series of parameters that are defined by a name:value, and separated by semicolons.
- Many different style parameters:



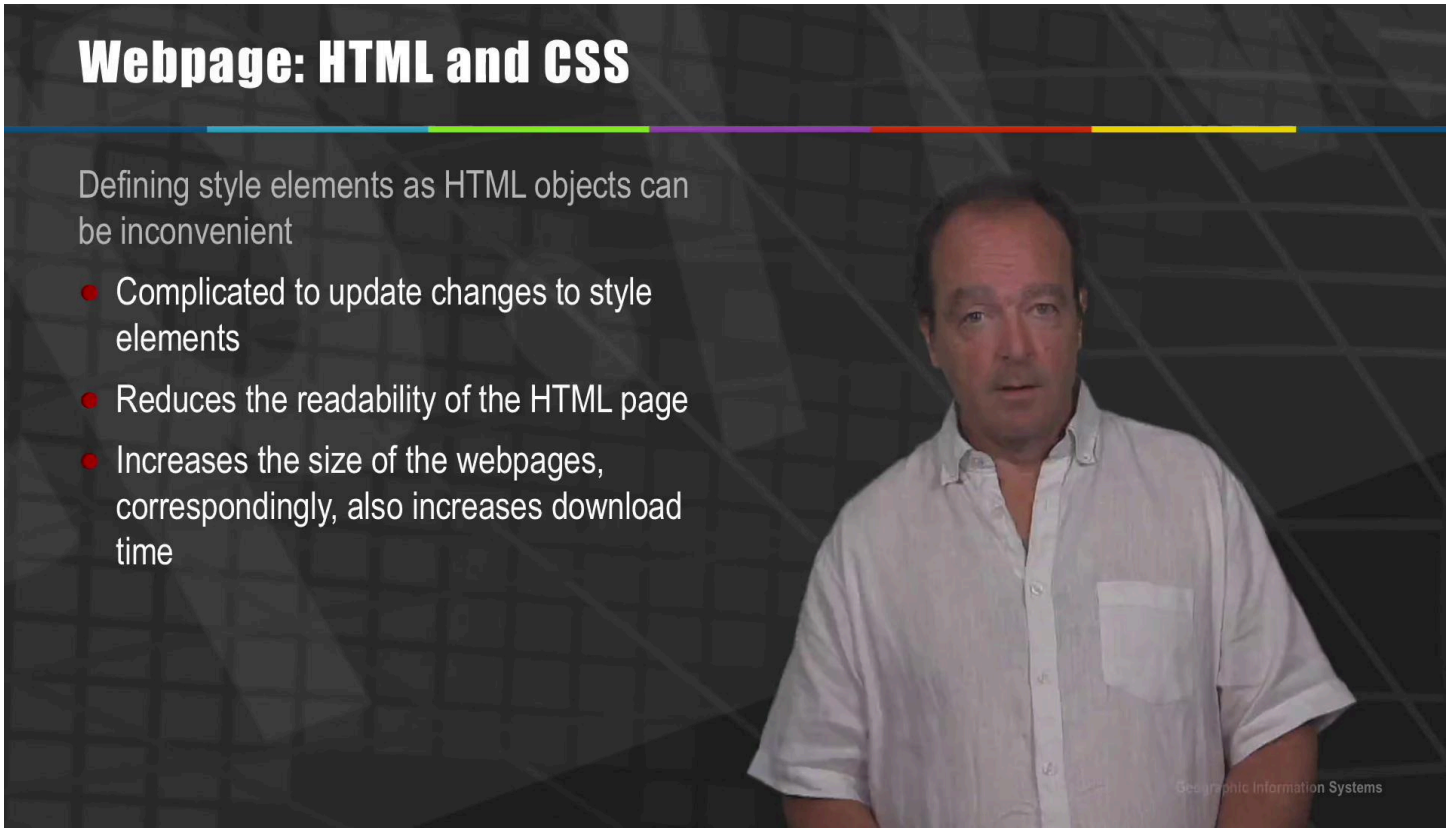
- For a detailed list of CSS elements:
  http://www.w3schools.com/css

We have seen here two examples of style applied to the content of two containers, of a SPAN container and a DIV container. Generally, HTML elements all have a style attribute which is actually composed of a string of characters which lists a series of parameters in a name:value form, separated by semicolons. There are a large number of style settings which can be used and we can, as we see here, go there to find details. We can find a fairly complete description again on the to w3schools site, in the CSS section.

Notes

Summary

17m 49s

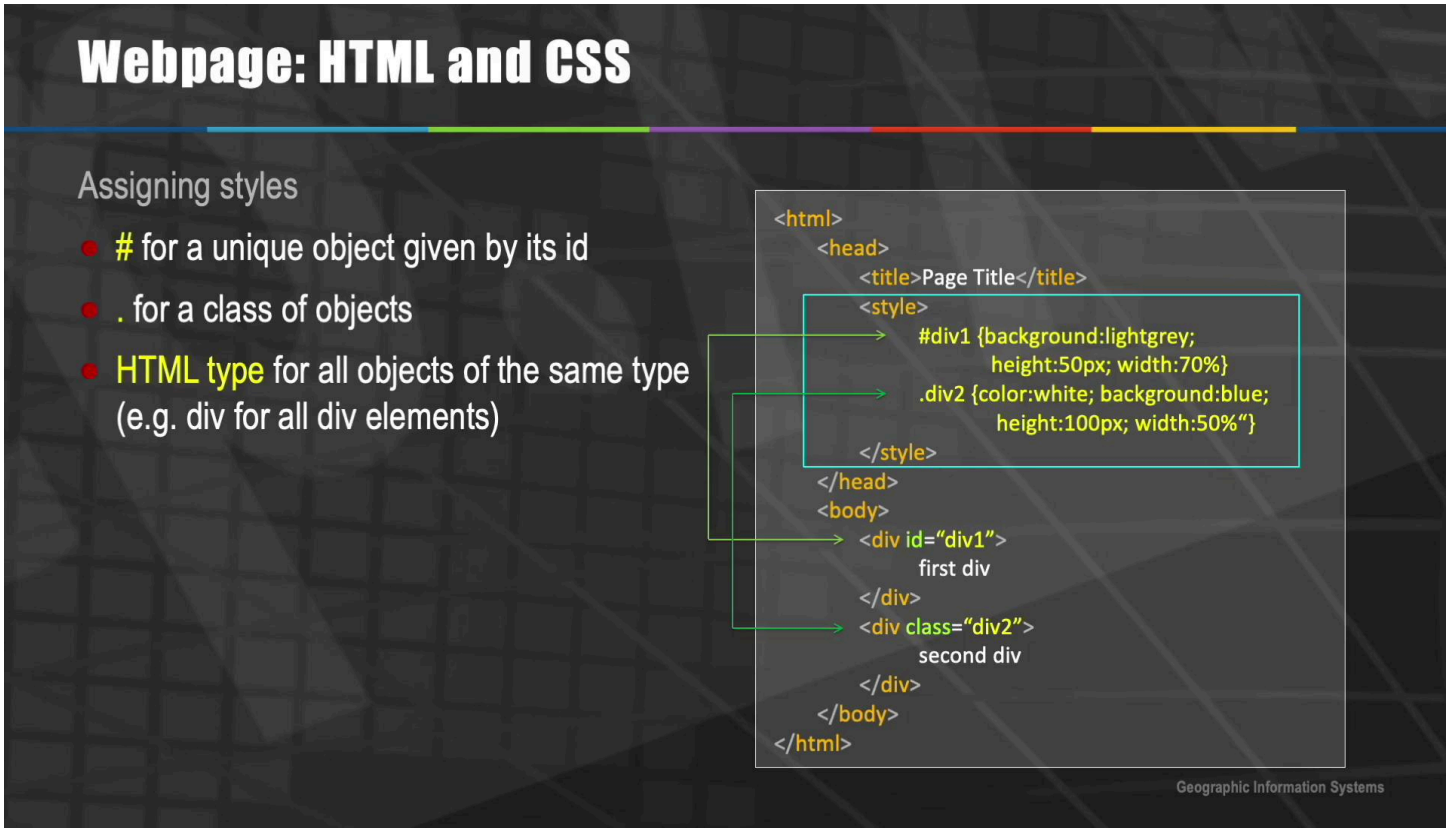


The method which consists of defining the style of HTML elements by parameters associated with these elements has a number of disadvantages: in the first place the fact that when we want to change style, to do an update, we are obliged to change the style specifically of each of the elements, what is a rather tedious task, the elements of style take more and more space, it actually makes it more difficult to read the web page and finally, if we repeat the same styles many times, we increase the size of the files, which can actually slow the loading times of web pages.

Notes

Summary

18m 35s

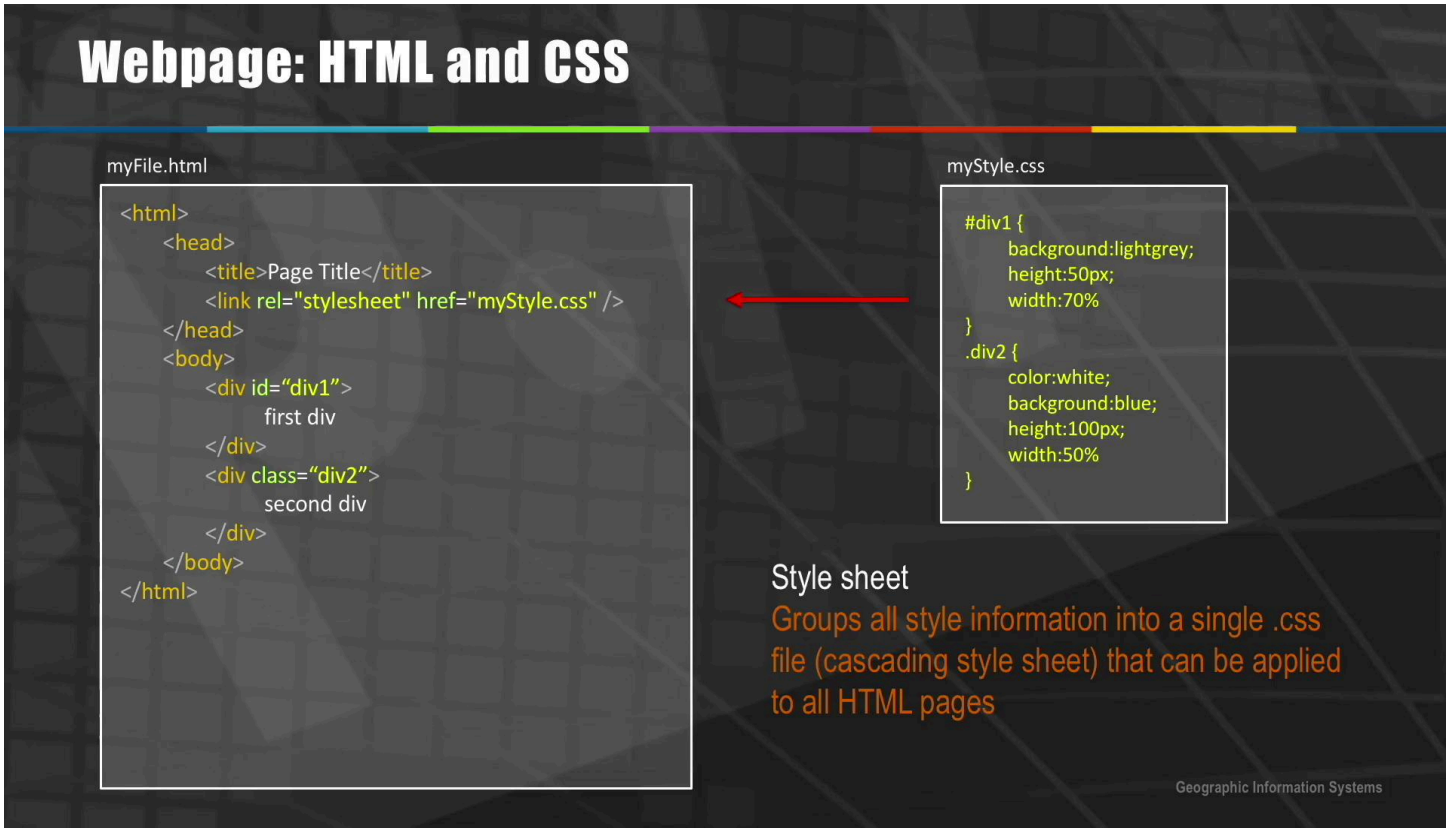


This is why we use alternative methods as illustrated in this example where we actually have just two containers, one above the other, first div, second div, one that occupies only 70% of the width of the page, with a height of 50 pixels and a light gray background and the second which has a dark blue background, a white text, a height of 100 pixels and which occupies half of the available width. So if we want to lighten up a bit the definition of styles, we will group the elements of stylization in a particular section of the page, a section that is characterized by style tags and in which, we will refer the different objects to which these styles apply, with the hashtag character to search for the element that carries this identifier, in this case here, div1, so the style applies to a single object, the point character to apply the style to a class of objects so we can, in the attributes of an object, define a class and we can have several objects that have the same class and so that we can stylize with a single style, and finally, the HTML type itself with a style that would then apply to all of the HTML elements of that type.

Notes

Summary

19m 19s

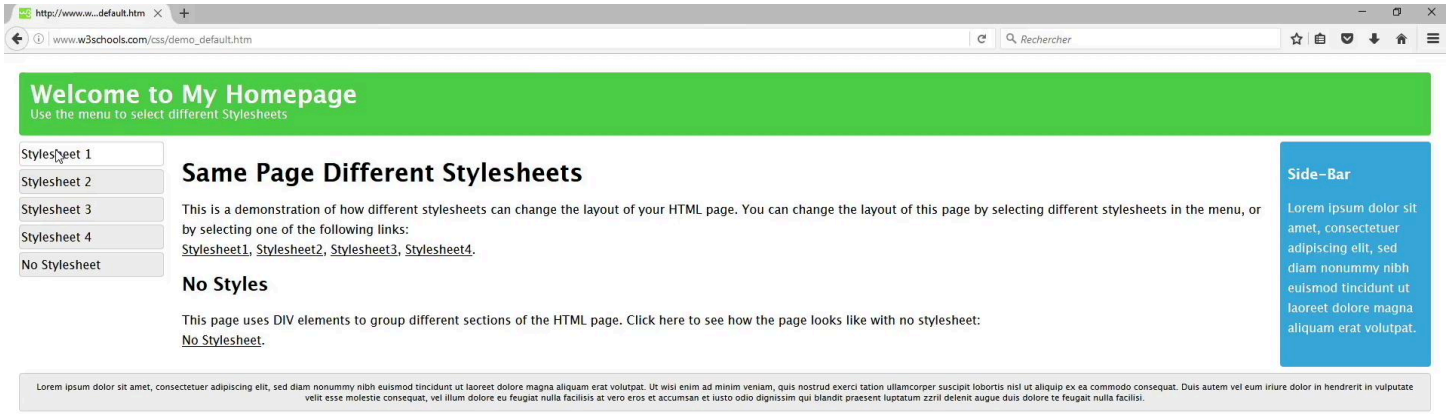


The next step of the rationalization of the style elements is to take the style section out of the HTML page header to group all the styles into a style page, so a.css page, which is then referenced in the HTML page with a link tag. This style sheet that we create gathers all the stylistic information in a single file, a.css file for Cascading Style Sheet and can be applied to all the HTML pages of a website.

Notes

Summary

20m 56s

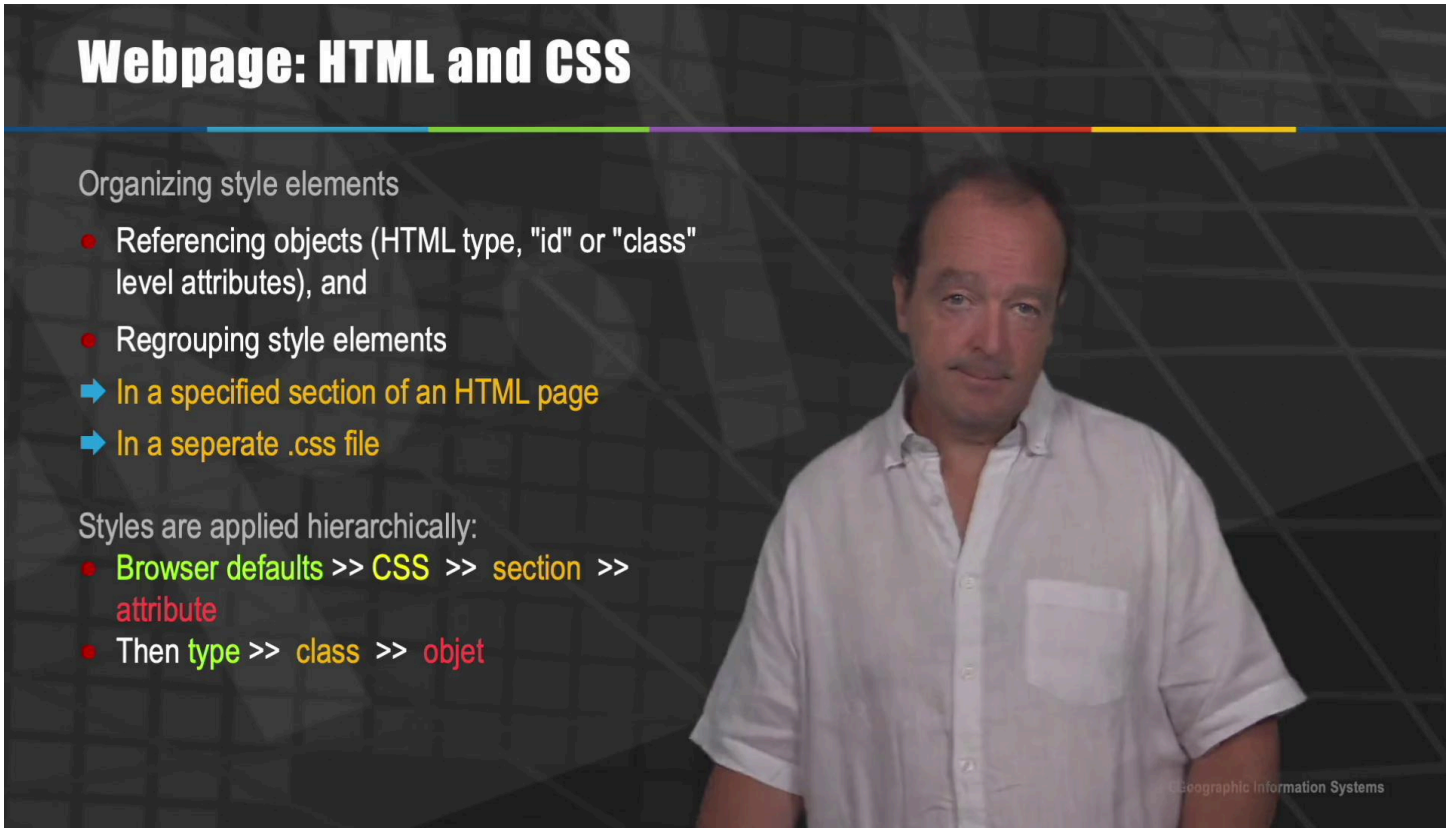


Here is an example of different types of stylization which are applied to one single page, to the same content, but with diversified shaping. This is the page as it appears without any style and we go back to square one.

Notes

Summary

21m 32s



We can see that in order to rationalize the style management in a page or a set of web pages, we will start by referencing the different HTML elements of web pages by their identifiers, classes or types and that we will group the style information, either in a particular section characterized by the style tag in the header of the HTML page, or in a separate.css file. With this approach, the style of an element can be defined in fact in several places and then the following hierarchy applies where we first go from the default style of the web page to the style defined in the CSS page, then in the section of the HTML page then as an attribute of the object itself. And similarly in the hierarchy of the typology, we will go first from the general type and then if a particular style is defined for a particular class, it is the class that will get the upper hand and in the case where the object itself has a definite style, it is the style of the object that will apply.

Notes

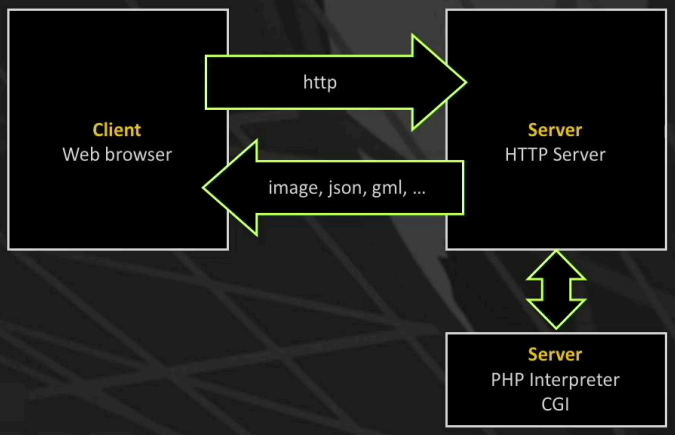
Summary

21m 55s

Web-GIS

Or on the server-side

- By .exe, .dll or scripts that are executed by the server to create dynamic HTML pages
  CGI - common gateway interface
- By codes that are executable at the server level
  PHP, java servlet, ASP – active server page, ColdFusion



Geographic Information Systems

We have seen the basic elements of construction and stylization of web pages, We still have to see how we can integrate mapping elements in these web pages and ensure that this mapping can be interactive. We have seen that the HTML language is rather limited and will not be enough for this type of more complex applications. The possibilities which are available to us, are to complete the HTML, which can be done either on the client side by adding executable code inside the HTML page, this code can take the form of Javascript, of vbscript or dhtml, by executable code to download and to use with an HTML page, then these are scripts in different programming languages, Python for example, activeX components, or Java applets. There is a wide variety of products available for this type of functions and by almost complete, specific application interfaces that we have to install on the client, which is commonly referred to as a plug-in. The possibilities can also be enriched on the server side by adding executables or DLL libraries, or even scripts that would be executed on the server and whose function is to create dynamically the HTML pages that are sent back to the client.

Notes

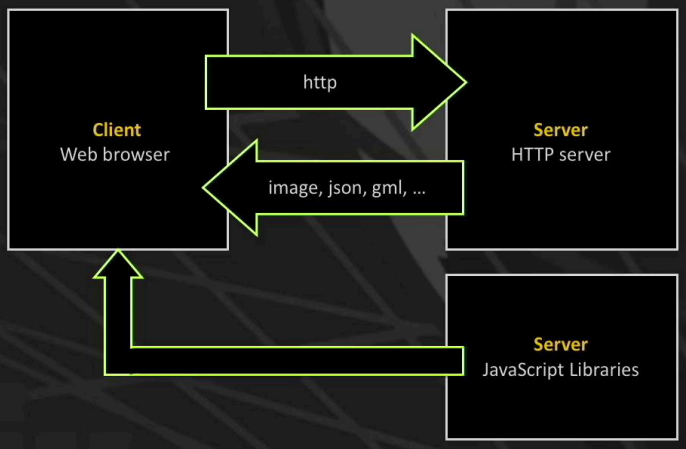
Summary

23m 21s

Web-GIS

HTML has its limitations; it is not sufficient for complex applications, but can be enhanced on the client-side

- By executable code that is incorporated into an HTML page HTML
  javascript, vbscript, dhtml
- By executable code that can be downloaded and used with an HTML page
  scripts, activeX, java applet
- By specific interface-applications that can be downloaded on the client-side
  plug-ins



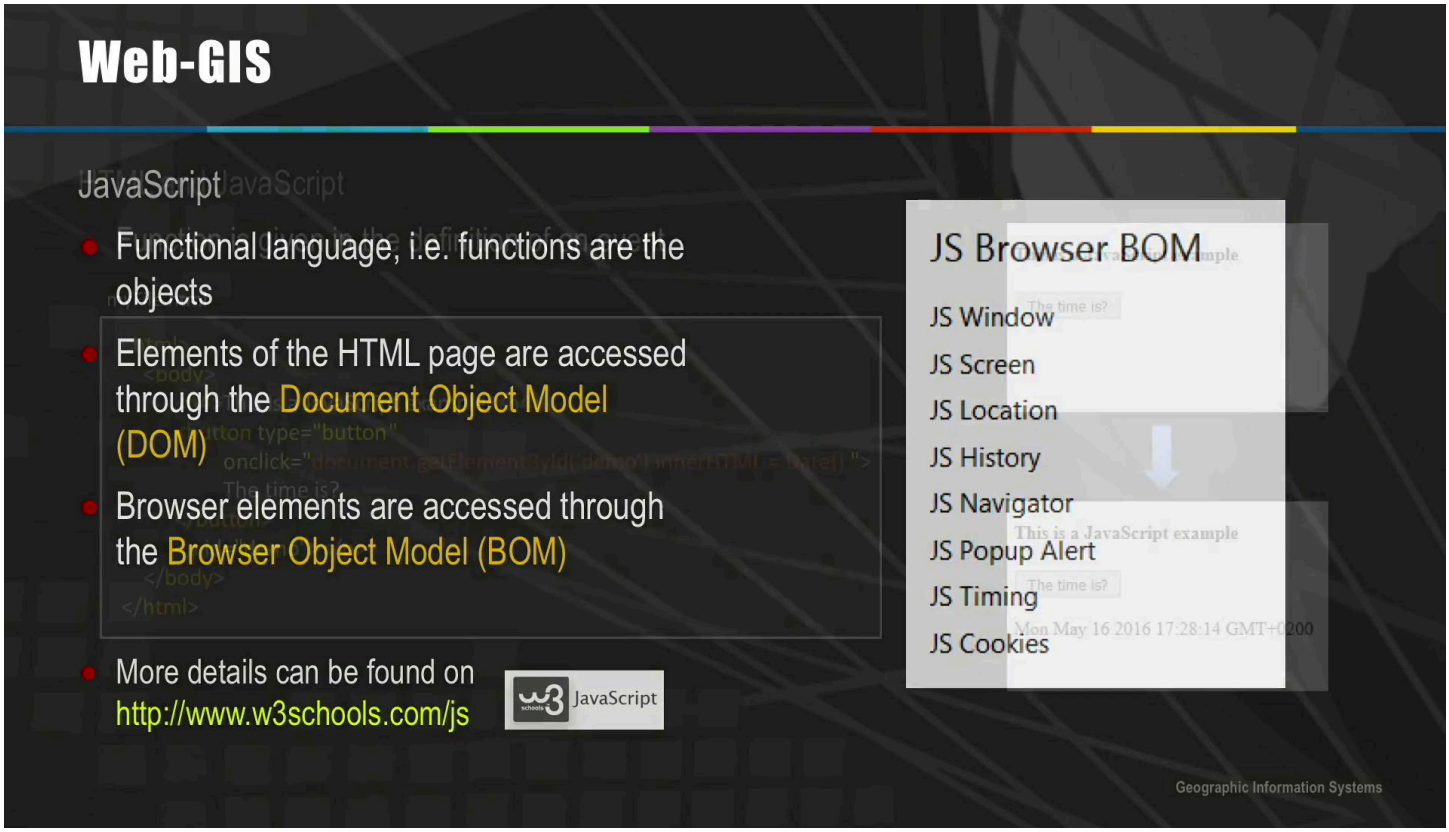
Geographic Information Systems

And there, we use a lot the common gateway interface, the CGI, or else by executable code which is integrated to the HTML page but is executed at the server level. The most common example used is that of PHP, so the PHP code that is in the page in this case a PHP page sent by the client to the server. The server must have a PHP interpreter to execute this code, make the HTML page and send it back to the customer. So we do not have time to review exhaustively all these different forms of HTML enrichment and we will concentrate in the rest of the course on the Javascript solution, so the Javascript executable code inserted into the HTML page and executed on the client side.

Notes

Summary

25m 05s

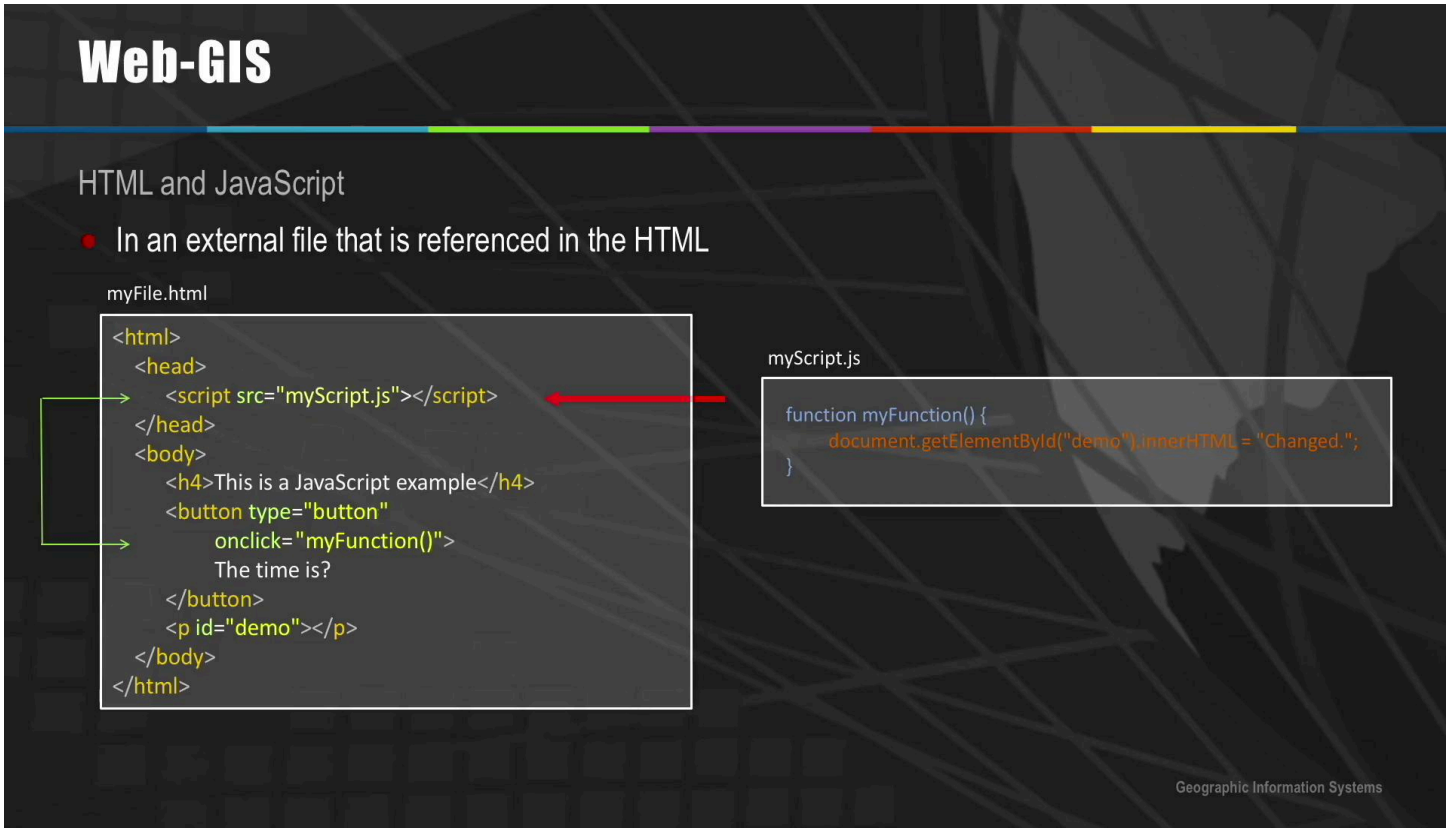


Javascript has, as one of its particularities, the fact of being a functional language, that is to say that the functions are objects. Here we have the example of a function of two parameters a and b which send back the product of these two variables. This function can be defined in a variable itself which would be called myFunction and this variable can be used to perform the function in the calculation of four times three. Javascript provides access to the elements of the HTML page via the Document Object Model, or DOM, which as we see on this illustration allows to access the HTML page which is the root element and then its components, the header and the body and in the header, the title, the title text or in the body, the various HTML elements that compose it. It is also possible to access the elements of the browser via the BOM, Browser Object Model, which enables to access the window, the browser history, the browser type, and so on. Again, if you want to deepen your knowledge on Javascript, I recommend you start by visiting the w3schools site which offers all sorts of exercises in this area.

Notes

Summary

26m 04s



We will see now how to integrate Javascript in an HTML page with a very simple example of a button in an HTML page and when we click on this button, we would like to see the day, the date, the time, etc. being displayed. The syntax of the HTML file is simple, so here, no header, just the body, with in the body, the title, the button and then a paragraph with the demo ID which must in fact receive the text of the date and time generated by the function. And we see that in the attributes of the button, we have an onclick property, so an event that will perform the function which is described in quotation marks. So from document, which is the DOM, we will search the object called demo, whose ID is demo, and then, we will assign to the HTML content of this object the Javascript date function that refers to the current date. As for the style elements, we have also the possibility to extract all the Javascript code of an HTML page or a series of HTML pages to group them in a js page, Javascript, and then to reference these different functions in the header with a script tag that points to the Javascript file whilst keeping at the level of the event, onclick on the button, the myFunction event which is defined in the myScript.js page. Here again, this process allows to simplify and facilitate the reading of the HTML page and group all the Javascript elements in the same place.

Notes

Summary

27m 36s

Web-GIS

Open source JavaScript libraries for GIS

- Openlayers
  http://openlayers.org/
- Leaflet
  http://leafletjs.com/
- Mapstraction
  http://mapstraction.com/
- Polymaps
  http://polymaps.org/

Geographic Information Systems

The integration of an interactive mapping in a web page using Javascript, will go through the use of Javascript SIG libraries.

Notes

Summary

29m 51s

Web-GIS

Open source GIS JavaScript libraries

- Openlayers
  http://openlayers.org/
- Leaflet
  http://leafletjs.com/
- Mapstraction
  http://mapstraction.com/
- Polymaps
  http://polymaps.org/
- Cesium (3D-Globes)
  https://cesiumjs.org/
- Modest Maps
- Jquery Geo
- …

Other GIS JavaScript libraries

- Google Maps
- Bing Maps
- MapQuest
- ESRI
- Via Michelin

Geographic Information Systems

There is a large number of these libraries including in particular Openlayers, Leaflet, the Mapstraction library, Polymaps, the Cesium library for three-dimensional representations, etc. Beyond these open source libraries, there is also a series of libraries which are generally free provided by the major producers that are Google, Microsoft, ESRI, etc. These libraries are free but they are not open source.

Notes

Summary

30m 10s

Example Application

Openlayers_Seychelles.html

```
1  <!doctype html>
2  <head>
3      <script src="http://openlayers.org/en/v3.10.1/build/ol.js" type="text/javascript"></script>
4      <link rel="stylesheet" href="http://openlayers.org/en/v3.10.1/css/ol.css" type="text/css">
5      <style>
6          html,body{
7              height:100%;
8              width:100%;
9          }
10         #map {
11             height:100%;
12             background-color: lightgrey;
13         }
14     </style>
15     <title>Openlayer Seychelles</title>
16 </head>
17 <body>
18     <div id ="map">
19     </div>
20 </body>
```

Map style elements

Create a Web-GIS for Mahé in the Seychelles with the Openlayers 3 library

- Define the page's style elements
- Define the map's style elements

Geographic Information Systems

In this last part of the course, we will demonstrate how to use one of these cartographic JavaScript libraries, in this case the Openlayers library, to create an HTML page which contains an interactive map. We start by creating a file entitled Openlayers_Seychelles.html in which we will find the basic elements, the header and the body with an element that we have not seen yet which is the !DOCTYPE html declaration which in fact substitutes itself for the html tags. Then, we add a title in the header then still in the header, two references, a first reference to the Openlayers library, and a link to a CSS file, a default style file which accompanies the Openlayers library. Then we add in the body of the page a container to which we assign the IDMap and we define some elements of style in a style section of the header with first for the entire HTML page and the whole of the body the desire to occupy all the available space, so 100% of the height and 100% of the width, and for the MAP element, also 100% of the height and then we give a light grey background color to be able to ensure that this HTML map element, this container, will occupy all the available space, something that we can verify by opening the file in a browser, in this case, Google Chrome.

Notes

Summary

30m 54s

Example Application

Openlayers_Seychelles.html

```
<!doctype html>
<head>
    <script src="http://openlayers.org/en/v3.10.1/build/ol.js" type="text/javascript"></script>
    <link rel="stylesheet" href="http://openlayers.org/en/v3.10.1/css/ol.css" type="text/css">
    <style>
        html,body{
            height:100%;
            width:100%;
        }
        #map {
            height:100%;
            background-color: lightgrey;
        }
    </style>
    <title>Openlayer Seychelles</title>
</head>
<body>
    <div id ="map">
    </div>

    <script>
        var map = new ol.Map({
            target: 'map',
            view: new ol.View({
              center: ol.proj.fromLonLat([55.47,  4.67]),
              zoom: 12
            })
        });

        // create and add the tile layer
        var osmlayer = new ol.layer.Tile({
          source: new ol.source.OSM()
        })
        map.addLayer(osmlayer);

    </script>
</body>
```

Define the OpenStreetMaps layer

Create a Web-GIS for Mahé in the Seychelles with the Openlayers 3 library

- Define the page's style elements
- Define the map's style elements
- Add a section for code
- Add the map
- Add the OpenStreetMap layer

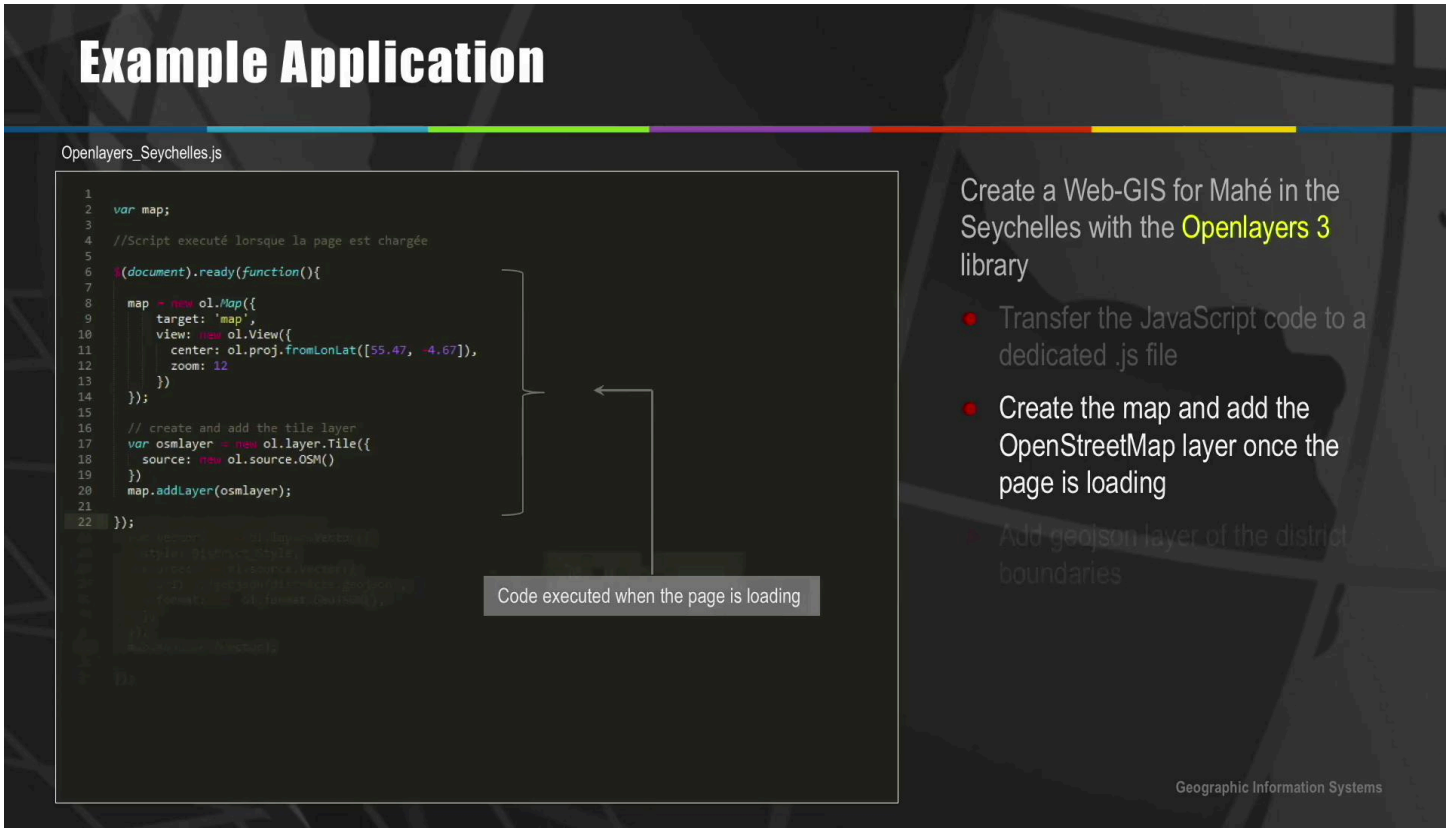
Geographic Information Systems

And we see that indeed, the component, the MAP container, occupies all the space available, with the exception of a small margin at the top and on the left. We then add a section to host the JavaScript code with script tags and in this script section, we will add the code that allows to create the map object, so we create a new OL.map object, so OL for openlayers, so a MAP object of the Openlayers library which we assign to the map variable and this object, in its properties, we say that the target that we aim at is the container called map, so we will put this cartographic element in the container, in the div which is the IDMap and then we define a view for this map, so we create a new view which is centered on the latitude, longitude point 55 47 and -4 67 which corresponds approximately to the center of the island of Mahé and this view, we also define, we can define a level of zoom as attribute, here the value 12. For something to be displayed in this map we will add a layer, in this case the OpenStreetMap layer. So we create a new tile layer with as OSM source, which is the OpenStreetMap source and we add this OSM layer to the map object which is the Openlayers map.

Notes

Summary

33m 00s



And we can see that if we open now this HTML file in a browser like Google Chrome, we have the OpenStreetMap map of the Seychelles of the island of Mahé, which appears in the web page. To add some content and interactivity to this map, we will now have to develop a little the JavaScript code and it is the reason why we are going to remove this script from the web page to host it in a JavaScript file that we will call Openlayers_Seychelles.js and we will reference this file in the header of the page. By taking this JavaScript code, we will start by separating the definition of the map variable of the use of this variable to assign it the new Openlayers map that we created, which allows to put the creation and addition code of the map of the OpenStreetMap layer in a function that will be executed only once when the page is loaded and once the page is loaded, we execute the code which adds the map, which adds the OpenStreetMap layer with the "document ready" key words, etc. Let's also note that the symbol $ refers to a JavaScript library which is called jQuery, a library that we needed to validly use this symbol, a library that had to be added in the HTML page, in the references.

Notes

Summary

34m 47s

Example Application

Openlayers_Seychelles.js

```
var map;

//Définition de quelques styles pour les couches vecteur
var District_Style = new ol.style.Style({
      fill: new ol.style.Fill({ color: 'rgba(200, 10, 10, 0.2)'}),
      stroke: new ol.style.Stroke({ color: 'rgba(255,0,0,1)', width: 1})
  });

//Script executé lorsque la page est chargée

$(document).ready(function(){

  map = new ol.Map({
      target: 'map',
      view: new ol.View({
        center: ol.proj.fromLonLat([55.47, -4.67]),
        zoom: 12
      })
  });

  // create and add the tile layer
  var osmlayer = new ol.layer.Tile({
    source: new ol.source.OSM()
  })
  map.addLayer(osmlayer);

  //Ajout de fichiers geojson
  var vector = new ol.layer.Vector({
    style: District_Style,
    source: new ol.source.Vector({
      url: './geojson/districts.geojson',
      format: new ol.format.GeoJSON(),
    })
  });
  map.addLayer(vector);

});
```

Create a Web-GIS for Mahé in the Seychelles with the Openlayers 3 library

- Transfer the JavaScript code to a dedicated .js file
- Create the map and add the OpenStreetMap layer when the page is loaded
- Add geojson layer of the district boundaries
- Define the symbology of the district layer

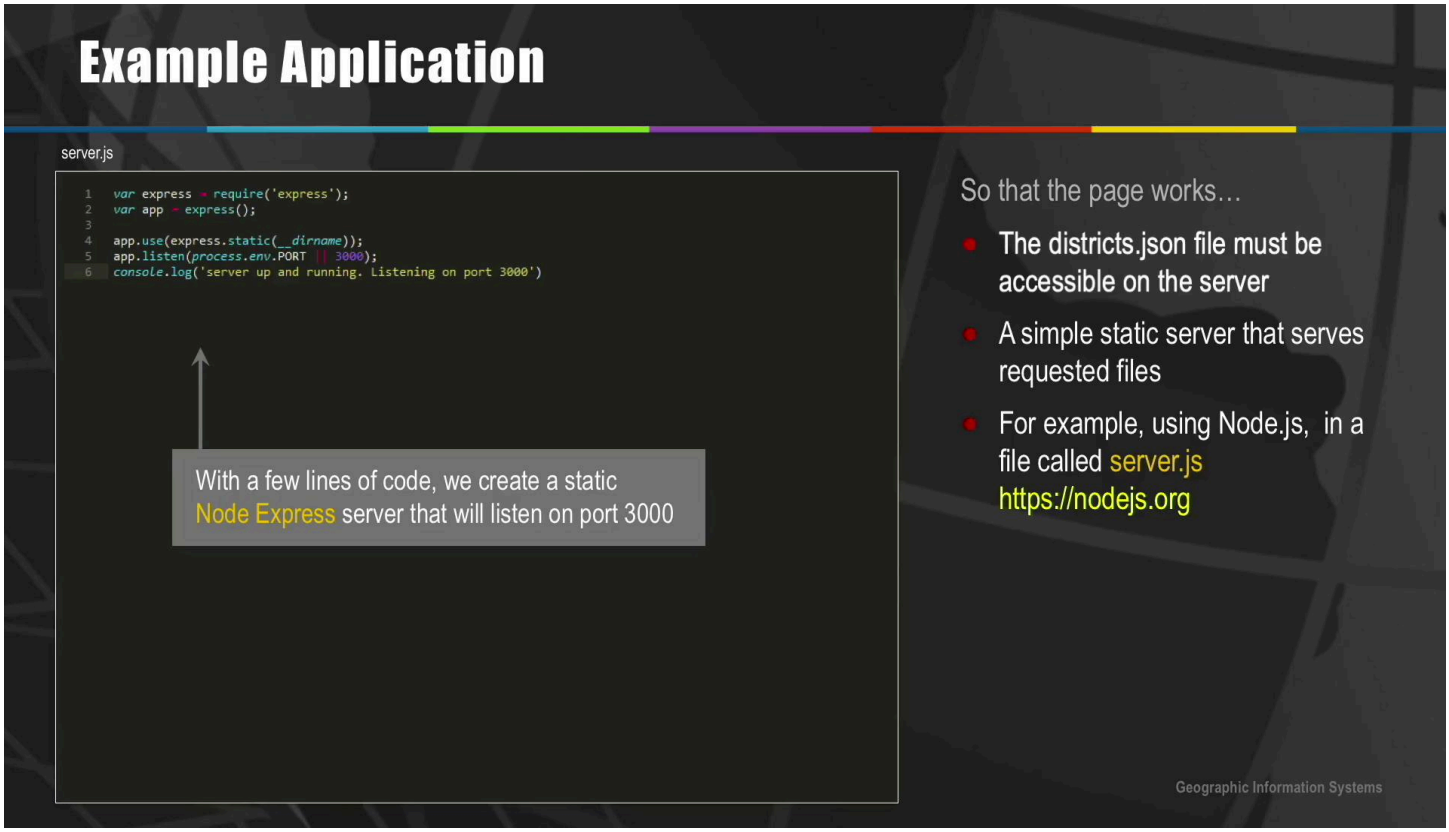
Geographic Information Systems

Let's suppose now that we want to enrich our map a little for example by adding the districts of the Seychelles which we stored here in a JSON file which is located, we see it in the URL, in a folder GeoJSON which is itself at the same level as our JavaScript file and our HTML page. So we create a vector layer, with a style that we will have to define, and we define this GeoJSON file as the source of this vector layer, and we add the layer to the map. We define here the symbology for the district layer by creating a new style composed of a filling, so a filling object created from a color and a transparency, and a stroke object which defines the line style for the polygon border also based on a color, a transparency, and a width. We can do the same thing to add the road layer, the hotel layer and then if we really want to add interactivity, we could write the code, we're not going to do it here because it gets a bit complicated but we could write the code which allows, when we click on a district for example or on a hotel to have a small pop-up window which opens in the web page And which gives the list of properties, of hotel attributes, of district, etc.

Notes

Summary

36m 29s

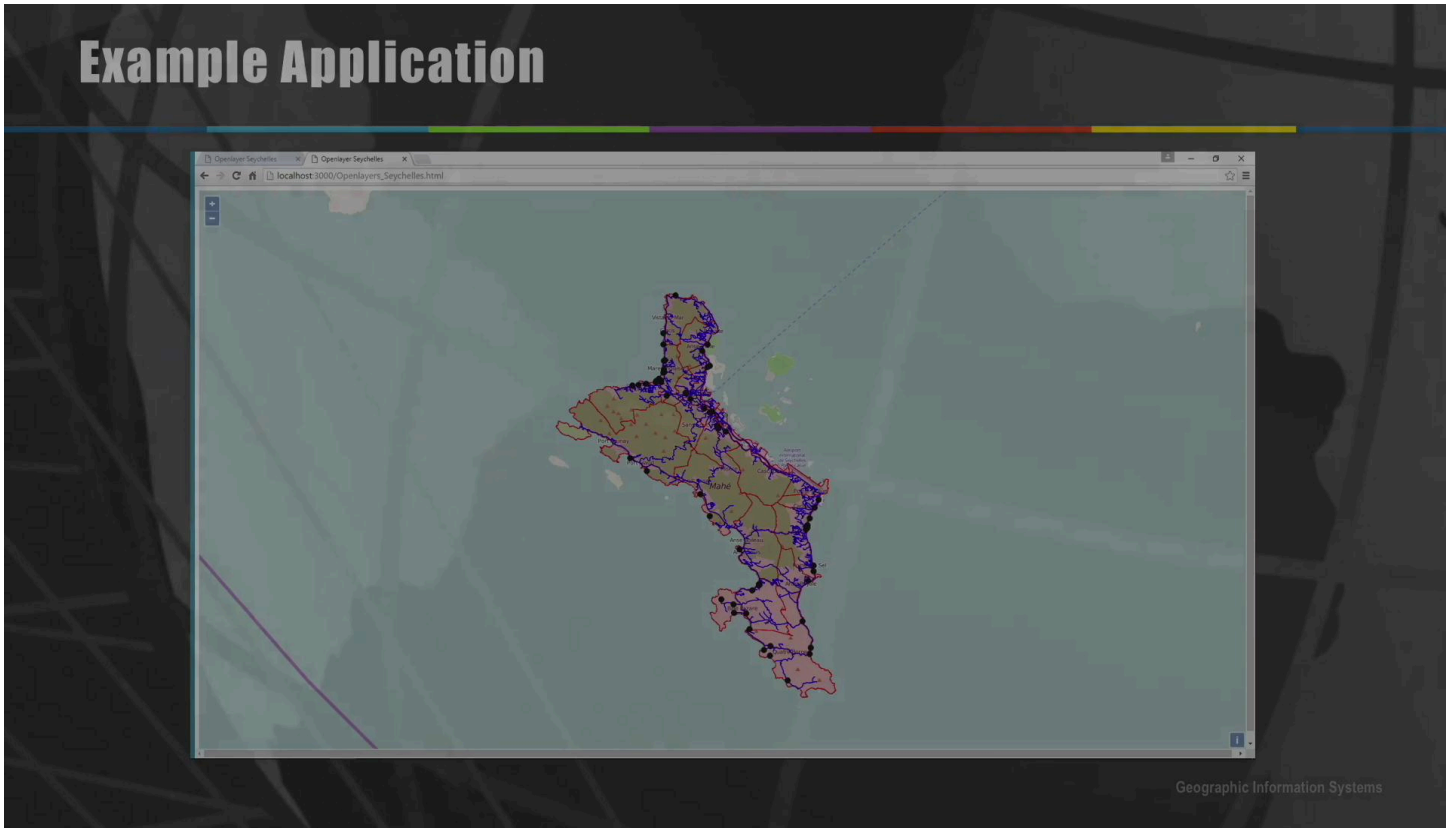


But for all this to work, it is still necessary to make these JSON files accessible, so we have to be able to expose them on a server, in this case we can simply use a simple static server which is not going to make manipulations and that will simply serve these files. It is a static server that we can create with the node.js library, so also JavaScript, in a file that we will call server.js and we see that in fact, in five lines of code, we create this server. A node variant called the Node Express which is a simplified form of the Node server which we defined in fact, the public folder where the files are located which will be made accessible, which is in this case the folder where this server.js file is located and the entire tree diagram that is located downstream. We will say that the server must listen to port 3000 and then send a message to the console to say that the server has started and that it has to listen to port 3000.

Notes

Summary

38m 08s



So if you take a quick look on the tree diagram of the files that we have created, so we have in the Openlayers_Samples folder these three JavaScript HTML files for the page web itself server.js for the server, the various modules of the node library which will enable to run this server, and in a GeoJSON file, we will find the three files containing the districts, the hotels and the roads of the Seychelles. We see here that the next step consists, in a terminal window, to go down in the tree diagram to this Openlayers_Samples folder and then send the command, node server.js which will launch the node server and we have the message that this server has started and that it listens to queries on port 3000. To load our web page, we have to open a web browser and type localhost as url since we are on a local machine, :3000 to search port 3000, so the node server that we have just started and then go to the Openlayers_Seychelles.html page which will execute the associated JavaScript code, which will itself go fetch the GeoJSON files and that will add them on the map, so that we find here the districts, roads and hotels. And as we said earlier, we can develop this code to add interactivity to this map in all kinds of forms.

Notes

Summary

39m 29s