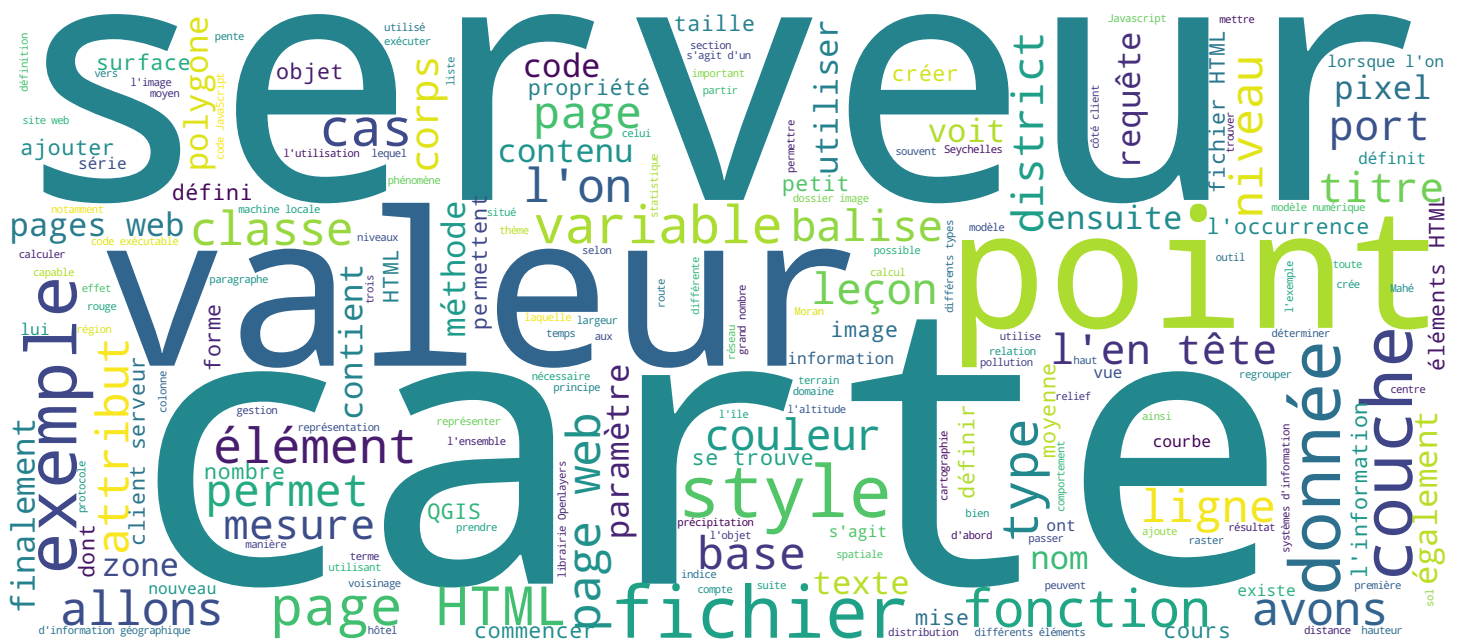


## Cartographie dynamique et interactive sur internet

# Introduction aux systèmes d'information géographique

Stéphane Joost, Marc Soutter, Fernand Kouamé, Amadou Sall



## Search MOOC



## Video



# Cartographie dynamique et interactive sur internet



## Objectifs de la leçon

- Examiner l'influence d'internet sur l'organisation et le fonctionnement des SIG
- Explorer le thème du Web SIG

Après cette leçon vous serez capables de

- D'expliquer le principe des architectures client-serveur
- De créer une page Web SIG simple

Introduction aux systèmes d'information géographique

Me revoilà donc et c'est avec grand plaisir que je vous retrouve pour cette leçon qui va porter sur certains aspects de la relation entre cartographie et Internet. Il s'agit d'un thème d'une grande actualité puisque de plus en plus souvent, les applications informatiques, et notamment les applications dans le monde des systèmes d'information géographique, s'appuient sur l'Internet pour proposer des solutions multi-plateformes, donc utilisables aussi bien sur des ordinateurs de bureau que des tablettes ou des smartphones et qui permettent d'utiliser différents types de système d'exploitation, que ce soit dans le monde Windows, le monde MacOS, Linux, Android, et j'en passe. L'objectif de cette leçon consiste donc à aborder, de manière générale et forcément un peu superficielle, le thème des relations entre systèmes d'information géographique et Internet, puis d'approfondir un peu le thème de l'intégration d'éléments de cartographie dynamique et interactive dans des pages web, ce que l'on désigne généralement par le terme de web SIG. Au terme de la leçon, vous devriez être en mesure de décrire les principes des architectures client-serveur, et vous devriez être capables de réaliser une page web simple contenant un élément de carte.

Notes

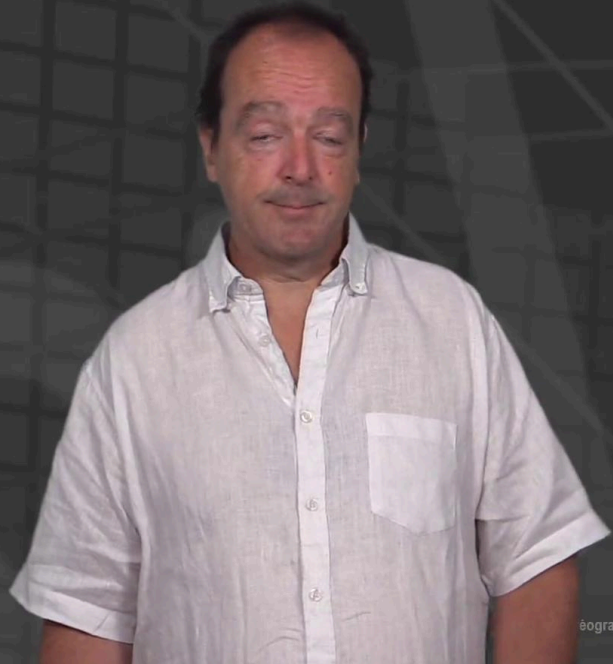
Summary



0m 18s

# Architecture client-serveur et SIG

- Un serveur est un **logiciel** ou une **machine** capable de recevoir de requêtes de clients et de leur répondre
- Leur objectif est de **partager** des **données** ou des **ressources** matérielles et/ou logicielles entre clients, dans une architecture dite client-serveur
- Les serveurs peuvent être des serveurs de **base de données**, de **fichiers**, de **courriel**, d'**impression**, des serveurs **web**, de **jeux**, d'**applications**, etc.



éographique

Nous aborderons donc successivement le thème de l'architecture client-serveur en relation avec les systèmes d'information géographique, puis nous nous concentrerons sur les éléments fondamentaux de la page web, que sont le langage HTML et le CSS, qui sont les prémices qui nous permettront d'aborder ensuite le thème de la cartographie dans les pages web, donc le thème du web SIG, et nous terminerons la leçon par un exemple d'application, donc de réalisation d'une page web qui contient un élément cartographique dynamique et interactif. Un serveur est donc un logiciel ou une machine qui est capable d'écouter le réseau et qui est capable de recevoir des requêtes émanant de différents clients et de répondre à ces requêtes. Donc l'objectif d'un serveur est de permettre le partage de données ou le partage de ressources matérielles ou logicielles entre plusieurs clients dans une architecture qualifiée d'architecture client-serveur. Il existe un très grand nombre de types de serveur différents : des serveurs de base de données, des serveurs simplement de fichiers, des serveurs de courriel, des serveurs web, des serveurs de jeux, etc.

Notes

Summary



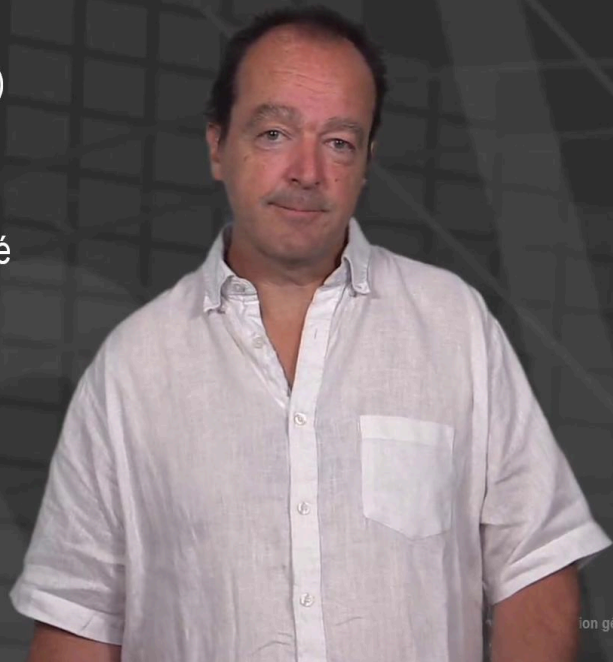
1m 43s



# Architecture client-serveur et SIG

## Port logiciel

- Du latin porta = porte (et non portus = port)
- Points d'entrée donnant accès au système d'exploitation d'un ordinateur
- Un port est identifié par un numéro exprimé sur 16 bits. Il existe donc un maximum de 65 536 ports ( $2^{16}$ ) par ordinateur
- Plusieurs serveurs peuvent tourner simultanément sur une machine en écoutant chacun un port spécifique



ion géographique

Les serveurs écoutent donc le trafic sur le réseau Internet afin d'identifier des requêtes qui leur sont adressées, et le portail qui leur permet d'écouter sur le réseau est ce qu'on appelle un port logiciel, qui est en fait le point d'accès au système d'exploitation d'un ordinateur. Les ports logiciels sont numérotés de 1 à 65'536 et sur une même machine, en fait, on peut avoir plusieurs logiciels serveurs qui fonctionnent simultanément, chacun écoutant ou travaillant avec un ou deux ports logiciels spécifiques, souvent un port pour le trafic entrant et un second port pour le trafic sortant.

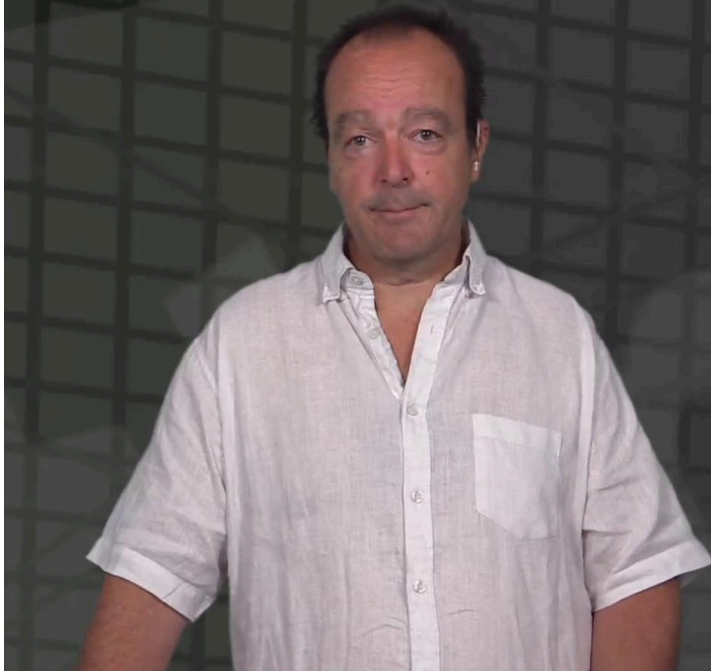
Notes

Summary



3m 05s

# Architecture client-serveur et SIG



## Exemples de numéros de ports usuels

- 20/21  
pour l'échange de fichiers via **FTP**
- 80  
pour la consultation d'un serveur **HTTP**
- 443  
pour la consultation d'un serveur **HTTPS**
- 5432  
Serveur de base de données **PostgreSQL**

Introduction aux systèmes d'information géographique

Comme exemple de numéros de port fréquemment utilisés, on peut citer les ports 20/21 qui sont utilisés pour les protocoles de transfert de fichier FTP, 20 pour le flux entrant 21 pour le flux sortant, le port 80 pour la consultation de pages web sur un serveur web HTTP, le port 443 dans le cas où ce serveur web serait sécurisé avec le protocole HTTPS, ou encore le port 5432 pour la connexion sur une base de données PostgreSQL.

Notes

Summary

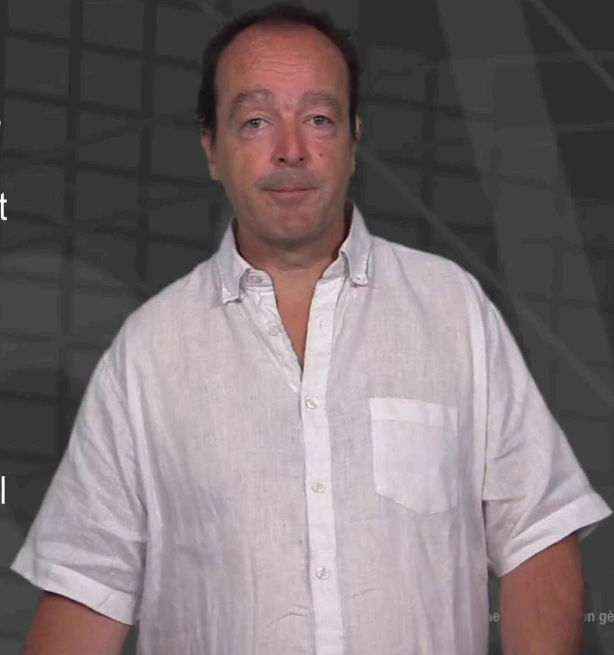


3m 49s

# Architecture client-serveur et SIG

localhost (ou hôte local)

- les logiciels serveur ne doivent pas forcément être hébergés sur des machines physiques ou virtuelles externes: une même machine physique peut parfaitement héberger le serveur et un ou plusieurs clients, exactement dans les mêmes conditions qu'un serveur externe, soit en communiquant par des **ports logiciels**
- Dans le domaine des réseaux informatiques, **localhost** est le nom habituel qui désigne une interface logique de l'ordinateur local



Notes

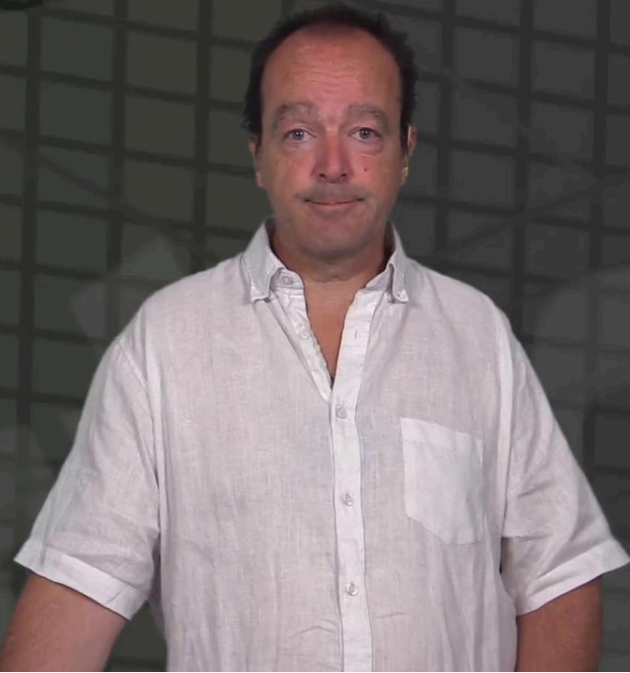
Les logiciels serveurs ne doivent pas nécessairement être hébergés sur une machine distante mais peuvent être hébergés sur la machine locale en même temps que les logiciels clients, ce qui est particulièrement utile si l'on fait du développement. Il faut simplement que les conditions de fonctionnement soient les mêmes que si les serveurs étaient sur une machine distante, à savoir que la communication entre le client et le serveur passe par les mêmes ports logiciels que si ces serveurs étaient sur une machine distante. La machine locale, dans une adresse URL, est désignée par le terme de "localhost" qui désigne en fait l'interface logique de cette machine locale.

Summary



4m 24s

# Architecture client-serveur et SIG



- Le nom **localhost** est associé à l'adresse IPv6 ::1 et à la plage d'adresses IPv4 127.0.0.0/8 (toutes les adresses comprises entre 127.0.0.1 et 127.255.255.255)
- Ainsi par exemple **http://localhost = http://127.0.0.1**
- Et l'adresse du port 3000 sur la machine locale sera **http://localhost:3000**

Introduction aux systèmes d'information géographique

L'adresse localhost est également associée à une série d'adresses IP. Dans le cas de IP version 4, c'est l'adresse 127.0.0.1 qui est la plus fréquemment utilisée, mais la série des adresses suivantes est également utilisable. Finalement, dans une adresse URL, si l'on veut spécifier un port particulier, on ajoute simplement deux points et le numéro de port pour viser ce port-là, donc si on veut taper sur le port 3000 d'une machine locale, l'URL serait du type `http://localhost:3000` Une architecture client-serveur est donc en général constituée par un logiciel serveur qui donc écoute le trafic sur le réseau, qui est capable d'identifier une requête qui lui est adressée et ensuite de renvoyer une réponse.

Notes

Summary



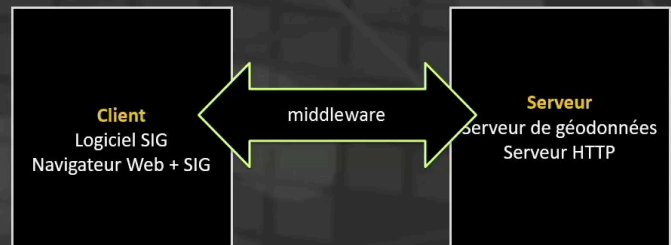
5m 10s



# Architecture client-serveur et SIG

Dans le contexte SIG

- Serveur
  - ➔ Serveur(s) de géodonnées
  - ➔ Serveur HTTP
- Client
  - ➔ Logiciel SIG
  - ➔ Navigateur Web + SIG
- Communication client-serveur
  - ➔ Middleware (intergiciel)



Introduction aux systèmes d'information géographique

De l'autre côté, on a un logiciel client qui lui est capable de formuler des requêtes, de les envoyer au serveur et puis de récupérer la réponse du serveur et de l'interpréter pour afficher l'information qui a été transmise. Finalement, les échanges entre client et serveur sont souvent gérés par un logiciel intermédiaire appelé intergiciel, ou encore middleware en anglais. Dans le contexte particulier des systèmes d'information géographique, le côté serveur est souvent simplement constitué d'un serveur de données. Il arrive cependant assez fréquemment que l'on ait deux serveurs en série, le premier étant destiné à gérer les requêtes et la fabrication des réponses, alors que le second se limite à la fourniture de données. Cette structure présente notamment des avantages du point de vue sécuritaire puisque l'accès aux données n'est autorisé que pour le serveur intermédiaire, qui lui peut gérer les droits d'accès des différents clients. Du côté client, on aura une interface utilisateur composée soit d'un logiciel SIG, QGIS, Manifold, Mapinfo, ArcGIS, etc., soit d'un navigateur web comprenant un composant cartographique, association que l'on désigne généralement par l'expression web SIG.

Notes

Summary



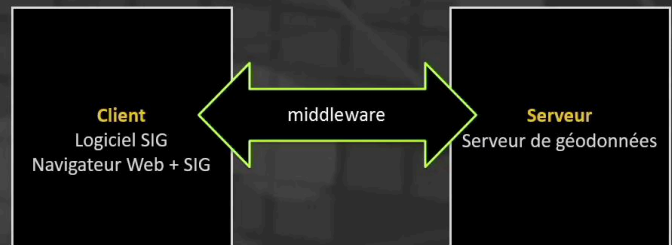


# Architecture client-serveur et SIG

## Exemples de middleware

### ODBC – Open Data Base Connectivity

- Transmet au serveur des requêtes SQL formulées au niveau du client
- Processus générique, valide pour toute base de données implémentant ODBC – souvent avec un **pilote ODBC** spécifique



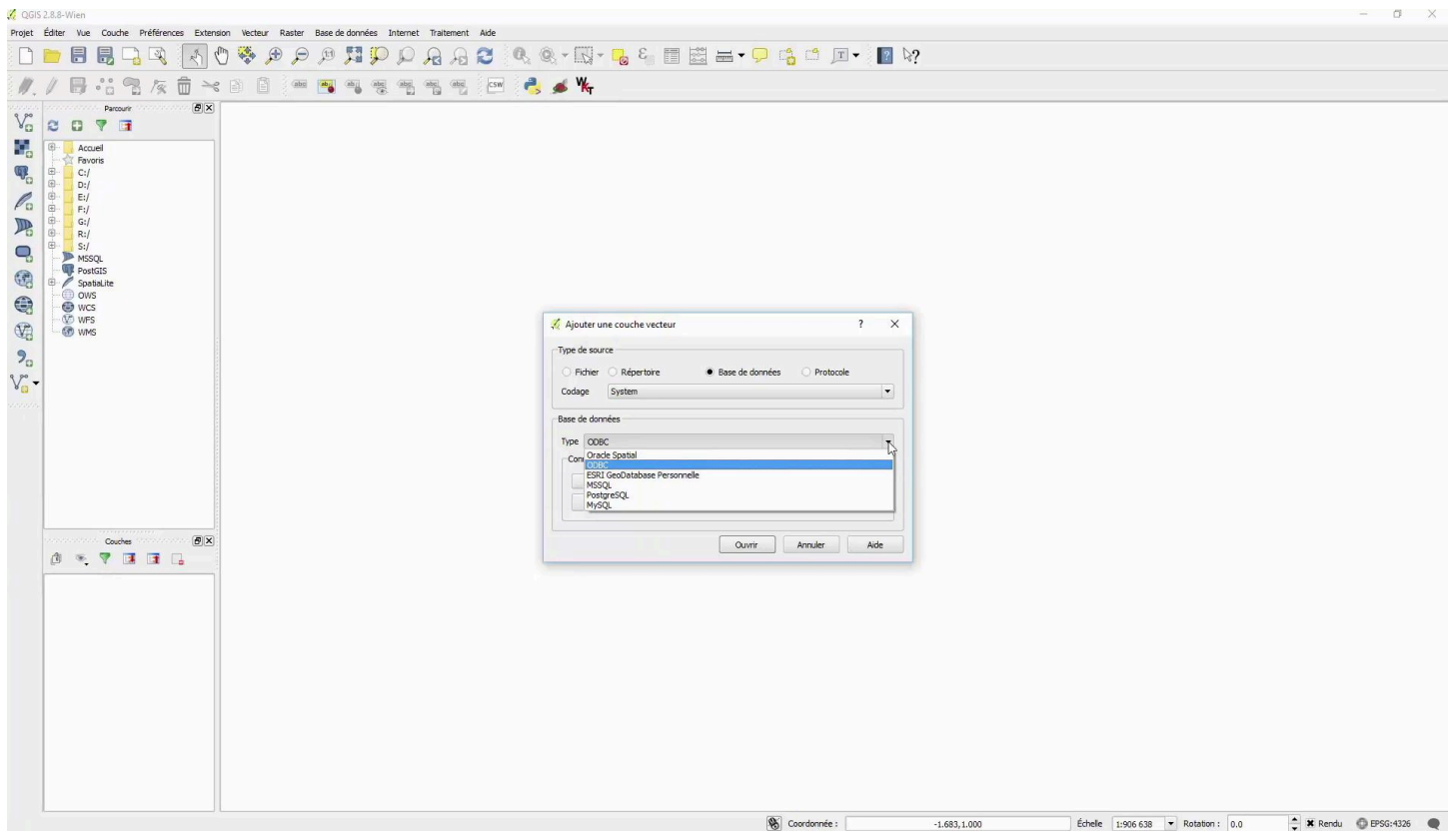
Introduction aux systèmes d'information géographique

Et à nouveau, on retrouvera divers protocoles de communication entre client et serveur. Le plus répandu de ces logiciels qui gèrent les communications entre client-serveur est le logiciel ODBC, pour Open Data Base Connectivity. Il s'agit d'un logiciel développé au départ par Microsoft pour le monde Windows, et repris par la suite par d'autres éditeurs pour les plateformes Unix/Linux et Java. Les requêtes de manipulation de données sont formulées en langage SQL dans le logiciel client, puis transmises au serveur de données par l'intermédiaire de ODBC. Le schéma de fonctionnement implique que les fabricants de logiciels de gestion de base de données ont dû développer des pilotes ODBC spécifiques pour chaque système de gestion de base données.

Notes

Summary





On voit ici dans le cas de QGIS que lorsque l'on ajoute par exemple une couche vectorielle, on a la possibilité d'aller rechercher cette couche dans une base de données, et les types de connexion base de données qui sont proposés comprennent différentes bases de données de différents fabricants, donc Oracle, MSSQL, Microsoft, Postgres, MySQL, mais également le pilote générique ODBC qui permet de se connecter à des bases de données en utilisant ce protocole particulier.

Notes

Summary

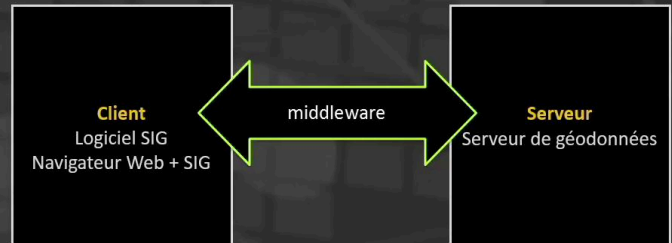


# Architecture client-serveur et SIG

## Exemples de middleware

### ODBC – Open Data Base Connectivity

- Transmet au serveur des requêtes SQL formulées au niveau du client
- Processus générique, valide pour toute base de données implémentant ODBC – souvent avec un **pilote ODBC** spécifique
- **OLEDB** – Object Linking and Embedding, Database successeur de ODBC basé sur les interfaces COM (Component Object Model)
- **ADO.NET** – composant d'accès aux données du framework .NET



Introduction aux systèmes d'information géographique

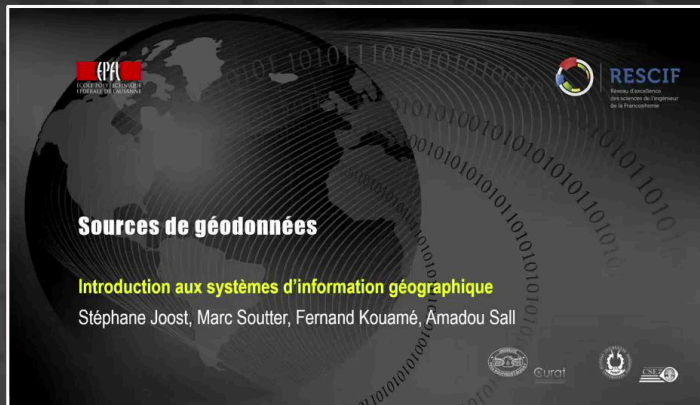
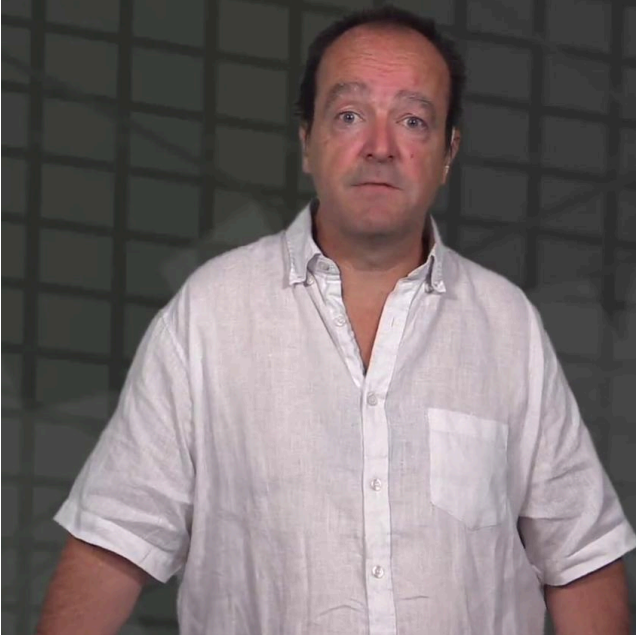
Deux autres exemples de middleware assez fréquemment rencontrés dans le monde Windows : OLEDB, pour Object Linking and Embedding Database, qui est en fait un successeur d'ODBC basé sur les interfaces COM, et ADO.NET, qui est le composant d'accès aux données du framework .NET de Microsoft.

Notes

Summary



# Architecture client-serveur et SIG



Introduction aux systèmes d'information géographique

Finalement, dans le cours consacré aux sources de géodonnées, nous avons déjà parlé un petit peu du protocole HTTP, en particulier pour interroger des géoservices, donc des serveurs WMS et WFS.

Notes

Summary



9m 06s



# Architecture client-serveur et SIG

## Cas particulier du protocole http

- Serveur
  - ➔ Serveur HTTP
  - ➔ Géoservices WMS, WFS
- Client
  - ➔ **Navigateur Web + SIG**
  - ➔ (logiciel SIG)
- Client-serveur
  - ➔ Protocole http
- Serveur-client
  - ➔ html, image, json, gml, etc.



Introduction aux systèmes d'information géographique

Dans le cas particulier du protocole HTTP, nous avons donc, du côté serveur, soit un serveur HTTP, soit des géoservices de type Web Map Service ou Web Feature Service. Côté client, un navigateur web avec un composant SIG ou, plus occasionnellement, le logiciel SIG, qui ne va généralement pas utiliser un serveur HTTP mais qui peut aller rechercher des tuiles dans des géoservices comme on l'a vu lors du cours sur les sources de géodonnées. La communication dans le sens client-serveur s'appuie donc sur le protocole HTTP, qui peut éventuellement comporter des paramètres, comme on l'avait vu pour définir les éléments que le serveur doit retourner et dans le sens client, on va trouver simplement un flux d'information sous la forme de HTML, d'images, de fichiers JSON, GML, etc. Dans la suite du cours, nous allons maintenant nous concentrer plus particulièrement sur le cas où le client est constitué d'un navigateur web qui comprend des fonctionnalités système d'information géographique.

Notes

Summary



# Page Web – html et css

## HTML – HyperText Markup Language

- Une page Web est un document **HTML** soit un simple fichier texte «annoté» afin de définir divers types de contenus (en-têtes, paragraphes, titres, liens, images, etc.)
- Les annotations (markup) sont formées par des balises (tag), i.e. des mots-clés placés entre chevrons et signalant le début `< >` et la fin `</ >` d'un élément de contenu ou élément **HTML**

```
<html>
  <head>
    <title> e-Atlas de l'Orontes </title>
  </head>
  <body>
    <p> Bienvenue sur l'e-Atlas de l'Orontes </p>
    
  </body>
</html>
```



Introduction aux systèmes d'information géographique

Avant d'aborder la question de l'intégration de composants géographiques dans une page web, il est nécessaire de rappeler un petit peu les fondements de l'organisation et de la construction d'une page web, donc le langage HTML pour HyperText Markup Language et le CSS pour les éléments de style. C'est donc l'objet de cette deuxième partie du cours, une revue accélérée un petit peu des principes de base de cette structuration de pages web. Une page web est donc avant tout un document HTML, donc un fichier.HTML, qui est un fichier texte annoté afin de définir différents types de contenu, l'entête, les pieds de page, les paragraphes, des titres, etc. Ces annotations qui portent le nom de Markup sont formées par des balises que l'on appelle des tags qui sont en fait des mots-clés placés entre chevrons et qui signalent le début et la fin d'un élément de contenu ou élément HTML. Dans l'exemple qu'on a ici sous les yeux, on voit que la balise principale est la balise HTML qui définit en fait la page HTML, elle-même subdivisée en un entête, avec la balise head qui contient un titre, avec la balise title, entête suivi d'une partie body qui est en fait le corps de la page, qui contient un paragraphe avec un texte, "Bienvenue sur l'e-Atlas de l'Orontes" et une image caractérisée par ses propriétés.

Notes

Summary



# Page Web – html et css

- L'en-tête contient des métadonnées et des références, et souvent le titre de la page `<title>`  
`</title>`
- La zone de contenu héberge les éléments **HTML** que l'on souhaite voir apparaître sur la page

```
<html>
  <head>
    <title> e-Atlas de l'Orontes </title>
  </head>
  <body>
    <p> Bienvenue sur l'e-Atlas de l'Orontes </p>
    
  </body>
</html>
```



Introduction aux systèmes d'information géographique

Le chemin d'accès à l'image a un texte de substitution au cas où l'image ne serait pas accessible et la taille en largeur de cette image. Les documents HTML devraient donc débuter et se terminer par des balises de début et de fin, balises HTML. Elles doivent contenir une section d'entête et une section de contenu, avec les balises head et body. Et finalement on note que la syntaxe du HTML n'est pas sensible à la casse, mais qu'en général on utilise plus volontiers des minuscules partout par confort de lecture. Et finalement, comme on l'a vu, l'entête contient des métadonnées, des références vers des bibliothèques, vers des ressources externes, souvent le titre de la page, et dans le corps on va placer les différents éléments de contenu, les différents éléments HTML que l'on souhaite voir apparaître sur la page elle-même.

Notes

Summary



# Page Web – html et css

## Quelques exemples de balises courantes

- `<h1>...<h6>` titres de niveau 1 à 6.
- `<p> </p>` paragraphes de texte
- `<em> </em>` marquer un élément de texte (par défaut en italique)
- `<strong> </strong>` marquer fortement un élément de texte (par défaut en gras)
- `<br>` saut de ligne
- `<hr>` ligne horizontale

```
<em>Emphasized text</em><br>  
<br>  
<strong>Strong text</strong><br>  
<br>  
<hr>
```

*Emphasized text*

**Strong text**

---

Introduction aux systèmes d'information géographique

Quelques exemples de balises courantes avec les balises de titre de niveau un à niveau six, la balise P pour les paragraphes, et puis une série de balises qui permettent de renforcer du texte en italique ou en gras, le saut de ligne, la ligne horizontale.

Notes

Summary



13m 20s



# Page Web – html et css

Quelques exemples de balises courantes

- `<a>` lien hypertexte
  - ➡ vers un site web
  - ➡ ou vers un élément de la page (défini par son identifiant id)

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

[Site Web de l'EPFL](http://www.epfl.ch)

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

This is the top  
[Back to top](#)

Introduction aux systèmes d'information géographique

Le lien HyperText avec la balise simplement A qui peut pointer soit vers un site web lorsque l'on donne comme paramètre href, l'adresse, l'url d'un site web, ici l'exemple du site web de l'École Polytechnique, ou vers un élément de la page situé plus haut ou plus bas qui est défini par son identifiant, ici on a l'exemple d'une phrase, "this is the top", qui est caractérisée par son ID qui s'intitule top et la référence renvoie à cet emplacement de la page, avec le caractère dièse que nous reverrons plus tard qui cible en fait un objet qui porte un ID top.

Notes

Summary



13m 38s

# Page Web – html et css

Quelques exemples de balises courantes

- `<ol>` `</ol>` listes ordonnées (liste numérotée)
- `<ul>` `</ul>` listes non ordonnées (liste à puces)
- `<li>` `</li>` éléments de listes

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

1. Coffee  
2. Tea  
3. Milk

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

50. Coffee  
51. Tea  
52. Milk

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

- Coffee
- Tea
- Milk

Introduction aux systèmes d'information géographique

Les balises permettant de définir des listes ordonnées avec la balise OL, non ordonnées avec la balise UL et la balise LI pour les éléments de listes.

Notes

Summary

14m 22s



# Page Web – html et css

Quelques exemples de balises courantes

- `<img>` Pour inclure une image
- Pas de balise de fin
- Attributs obligatoires
  - `src` le chemin d'accès à la source, sous forme d'URL ou de chemin local (relatif)
  - `alt` un texte alternatif affiché dans le cas où la source n'est pas accessible
- La valeur des attributs doit être entre guillemets, simples ou doubles

```

```



La balise image que nous avons déjà vue, avec pour particularité le fait que cette balise ne nécessite pas de balise de fin et comme autre particularité que ses deux attributs, la source et le texte de substitution, sont des attributs obligatoires. Et on note au passage que la valeur des attributs est caractérisée par un texte pris entre guillemets ou apostrophes.

Notes

Summary



14m 33s

# Page Web – html et css

## Quelques exemples de balises courantes

- `<img>` Pour inclure une image
- Pas de balise de fin
- Attributs obligatoires
  - `src` le chemin d'accès à la source, sous forme d'URL ou de chemin local (relatif)
  - `alt` un texte alternatif affiché dans le cas où la source n'est pas accessible
- La valeur des attributs doit être entre guillemets, simples ou doubles

```

```

## Exemples de chemins d'accès pour une source qui se trouve:

- dans le même dossier que le fichier HTML  
`src = "sources.jpg"` ou `src = "./sources.jpg"`
- dans un dossier images placé au même niveau que le fichier HTML  
`src = "../images/sources.jpg"`
- dossier images au même niveau que le dossier HTML contenant le fichier HTML  
`src = "../../images/sources.jpg"`
- hébergé sur un site Web  
`src = "http://www.mysite.ch/OrontesImg/sources.jpg"`

Introduction aux systèmes d'information géographique

La définition des chemins d'accès suit les règles standards. Ainsi par exemple dans le cas d'une source qui se trouve dans le même dossier que le fichier HTML, la source sera simplement constituée par le nom du fichier ou éventuellement par la syntaxe: (point) qui désigne le dossier dans lequel on se trouve, le nom du fichier. Si on a un dossier image qui est placé au même niveau que le fichier HTML sur lequel on travaille, le lien vers la source prendra la forme ./dossier image/nom du fichier. Il peut arriver que le dossier image soit au même niveau que le dossier HTML qui contient le fichier HTML sur lequel on travaille, et dans ce cas pour remonter d'un cran dans la hiérarchie, on va avoir une syntaxe du type ../dossier image/fichier source.jpeg Et finalement dans le cas où l'image serait hébergée sur un site web distant, on aurait simplement pour la source l'url de cette image.

Notes

Summary



15m 01s



# Page Web – html et css

Quelques exemples de balises courantes

- `<table>` `</table>` définissent une table composée de
- `<tr>` `</tr>` lignes (table row), elles-mêmes composées de
- `<td>` `</td>` cellules (table data)

```
<table>
<tr>
  <th>Month</th>
  <th>Savings</th>
</tr>
<tr>
  <td>January</td>
  <td>$100</td>
</tr>
<tr>
  <td>February</td>
  <td>$80</td>
</tr>
</table>
```

Month	Savings
January	\$100
February	\$80

- Pour une liste complète  
<http://www.w3schools.com/html>



Introduction aux systèmes d'information géographique

La balise table qui définit une table composée donc de lignes, avec la balise TR pour row, chaque ligne étant composée de cellules décrites par la balise TD. On peut trouver une liste complète des balises de HTML sur le site w3schools, dans la partie HTML de ce site.

Notes

Summary



16m 12s

# Page Web – html et css

- Les éléments neutres `<div>` `</div>` et `<span>` `</span>` sont simplement des conteneurs pour d'autres éléments HTML et servent de ce fait à structurer et styliser le document HTML.
- L'élément `div` est encadré par des sauts de ligne (élément bloc), alors que l'élément `span` ne possède pas de formatage par défaut (élément en ligne)

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

My Important Heading

```
<p>This is some text.</p>
```

This is some text.

```
<div style="color:#0000FF;background:lightgrey">
```

```
  <h3>This is a heading in a div element</h3>
```

This is some text in a div element

```
  <p>This is some text in a div element</p>
```

This is some text in a div element

```
</div>
```

This is some text.

```
<p>This is some text.</p>
```

Introduction aux systèmes d'information géographique

Pour terminer encore, les éléments neutres constitués par les balises `DIV` et `SPAN` qui sont en fait simplement des conteneurs pour accueillir d'autres éléments HTML et qui sont principalement utilisés pour structurer et styliser un document HTML. L'élément `DIV` est lui encadré par des sauts de ligne, donc il s'agit d'un élément de bloc, et chaque élément suivant va se retrouver à la ligne suivante, alors que l'élément `SPAN` lui est un élément en ligne et peut s'appliquer à une section de ligne, avec ici l'exemple, du moins important, qui est stylisé en rouge à l'aide d'un `SPAN` dans la phrase titre et puis le second exemple, un élément `DIV` qui est coloré en gris, un arrière fond en gris et puis un texte en bleu, style qui est appliqué au contenu de ce conteneur, donc le titre de niveau `h3` et le paragraphe qui suit.

Notes

Summary



16m 39s

# Page Web – html et css

## HTML & CSS – Cascading Style Sheet

- Les éléments HTML ont un attribut style, soit un string composé d'une série de paramètres sous la forme **nom:valeur**, séparés par des point-virgules
- Paramètres de style nombreux et diversifiés

- [Color](#)
- [Background and Borders](#)
- [Basic Box](#)
- [Flexible Box](#)
- [Text](#)
- [Text Decoration](#)
- [Fonts](#)
- [Writing Modes](#)

- [Table](#)
- [Lists and Counters](#)
- [Animation](#)
- [Transform](#)
- [Transition](#)
- [Basic User Interface](#)
- [Multi-column](#)

- [Paged](#)
- [Genera](#)
- [Filter E](#)
- [Image/](#)
- [Maskin](#)
- [Speech](#)
- [Marque](#)

### Color Properties

Property	Description	CSS
color	Sets the color of text	1
opacity	Sets the opacity level for an element	3

### Background and Border Properties

Property	Description	CSS
background	A shorthand property for setting all the background properties in one declaration	1
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	1
background-blend-mode	Specifies the blending mode of each background layer (color/image)	3
background-color	Specifies the background color of an element	1
background-image	Specifies one or more background images for an element	1
background-position	Specifies the position of a background image	1
background-repeat	Sets how a background image will be repeated	1
background-clip	Specifies the painting area of the background	3
background-origin	Specifies where the background image(s) is/are positioned	3
background-size	Specifies the size of the background image(s)	3
border	Sets all the border properties in one declaration	1
border-bottom	Sets all the bottom border properties in one declaration	1

- Pour un inventaire détaillé  
<http://www.w3schools.com/css>



Nous avons vu là deux exemples de style appliqués aux contenus de deux conteneurs, d'un conteneur SPAN et d'un conteneur DIV. De manière générale, les éléments HTML ont tous un attribut style qui est composé en fait d'une chaîne de caractères qui liste une série de paramètres sous une forme nom:valeur séparé par des point-virgules. Il existe un très grand nombre de paramètres de style que l'on peut utiliser et l'on peut, comme on le voit ici, aller piocher là-dedans pour trouver des détails. On trouvera une description assez complète de nouveau sur le site w3schools, dans la rubrique CSS.

### Notes

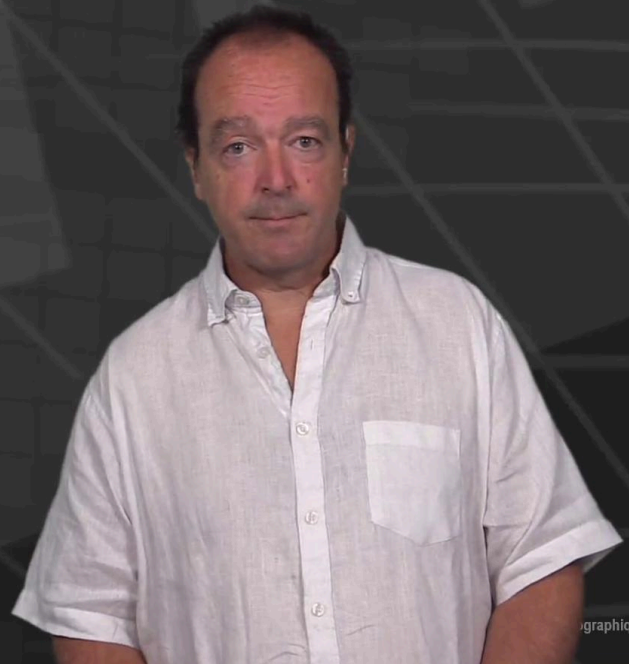
### Summary



# Page Web – html et css

La définition des éléments de style comme attributs des éléments HTML a des inconvénients

- Lourdeur de la mise à jour lors du changement d'un élément stylistique
- Dégradation de la lisibilité de la page HTML
- Augmentation de la taille des pages et donc du temps de téléchargement



La méthode qui consiste à définir le style des éléments HTML par des paramètres associés à ces éléments possède un certain nombre d'inconvénients : en premier lieu le fait que lorsque l'on veut faire un changement de style, une mise à jour, on est obligé d'aller changer le style spécifiquement de chacun des éléments, ce qui est une tâche assez fastidieuse, les éléments de style prennent de plus en plus de place, ça alourdit en fait la lisibilité de la page web et finalement, si on répète les mêmes styles de nombreuses fois, en fait on augmente la taille des fichiers, ce qui peut ralentir en fait les temps de chargement des pages web.

Notes

Summary



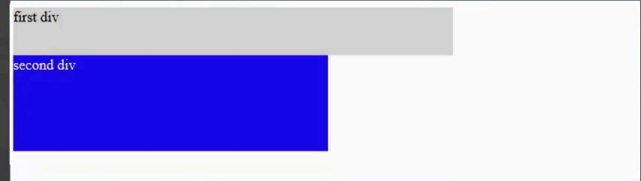
18m 35s



# Page Web – html et css

Éléments de style définis au niveau des objets

```
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <div style="background:lightgrey; height:50px;
      width:70%">
      first div
    </div>
    <div style="color:white; background:blue;
      height:100px; width:50%">
      second div
    </div>
  </body>
</html>
```



Introduction aux systèmes d'information géographique

C'est pour cette raison que l'on utilise des méthodes alternatives comme on l'illustre au travers de cet exemple où l'on a en fait simplement deux conteneurs, l'un en-dessus de l'autre, first div, second div, l'un qui n'occupe que 70% de la largeur de la page, avec une hauteur de 50 pixels et un fond gris clair et le second qui a un fond bleu foncé, un texte de couleur blanche, une hauteur de 100 pixels et qui occupe la moitié de la largeur disponible.

Notes

Summary



# Page Web – html et css

## Assignation des styles

- # pour l'objet unique décrit par son id
- . pour une classe d'objets
- **Type HTML** pour tous les objets d'un même type (p.ex. div pour tous les div)

```
<html>
  <head>
    <title>Page Title</title>
    <style>
      #div1 {background:lightgrey;
        height:50px; width:70%}
      .div2 {color:white; background:blue;
        height:100px; width:50%"}
    </style>
  </head>
  <body>
    <div id="div1">
      first div
    </div>
    <div class="div2">
      second div
    </div>
  </body>
</html>
```

Introduction aux systèmes d'information géographique

Alors si l'on veut alléger un petit peu la définition des styles, on va regrouper en fait les éléments de stylisation dans une section particulière de la page, section que l'on caractérise par les balises style et dans laquelle, on va référencer en fait les différents objets auxquels ces styles s'appliquent, avec le caractère dièse pour rechercher l'élément qui porte cet identifiant, en l'occurrence ici, div1, donc le style s'applique à un seul objet, le caractère point pour appliquer le style à une classe d'objets donc on peut, dans les attributs d'un objet, définir une classe et on peut avoir plusieurs objets qui ont la même classe et donc, que l'on peut styliser avec un seul style, et finalement, le type HTML lui-même avec un style qui s'appliquerait alors à l'ensemble des éléments HTML de ce type-là.

Notes

Summary



# Page Web – html et css

myFile.html

```
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="myStyle.css" />
  </head>
  <body>
    <div id="div1">
      first div
    </div>
    <div class="div2">
      second div
    </div>
  </body>
</html>
```

myStyle.css

```
#div1 {
  background:lightgrey;
  height:50px;
  width:70%
}
.div2 {
  color:white;
  background:blue;
  height:100px;
  width:50%
}
```

## Feuille de style

Regroupement de toutes les informations stylistiques dans un fichier .css (cascading style sheet) qui peut être appliqué à l'ensemble des pages HTML d'un site Web

Introduction aux systèmes d'information géographique

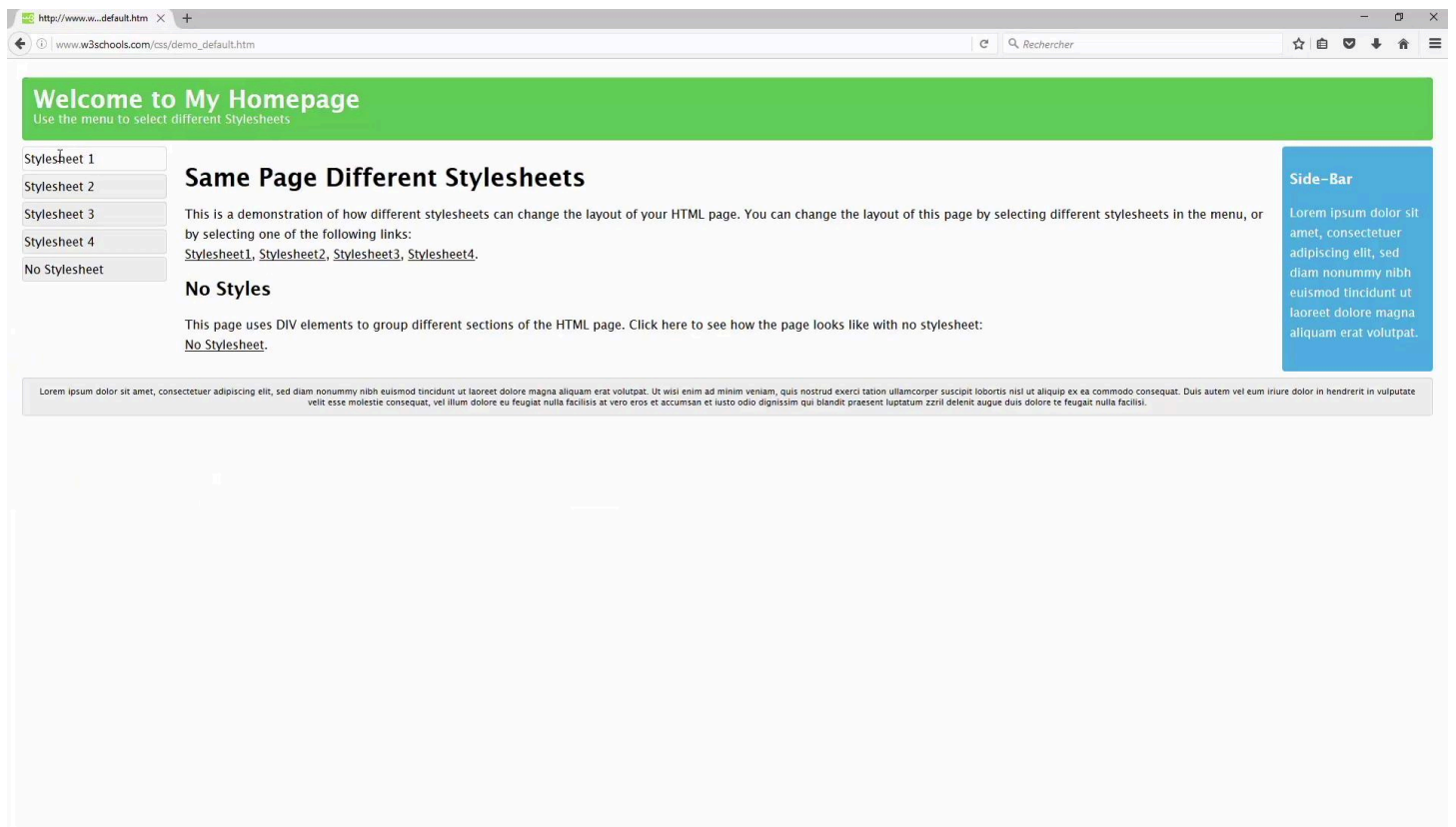
L'étape suivante de la rationalisation des éléments de style consiste à sortir la section style de l'en-tête de la page HTML pour regrouper tous les styles dans une page de style, donc une page.css, que l'on référence ensuite dans la page HTML avec une balise de type link. Cette feuille de style que l'on crée regroupe toutes les informations stylistiques dans un seul fichier, un fichier.css, pour Cascading Style Sheet, et peut être appliquée à l'ensemble des pages HTML d'un site web.

Notes

Summary



20m 56s



On a ici un exemple de différents types de stylisation qui sont appliqués à une seule et même page, à un même contenu, mais avec des mises en formes diversifiées. Ca, c'est la page telle qu'elle se présente sans style aucun et on retourne à la case départ.

Notes

Summary



21m 32s

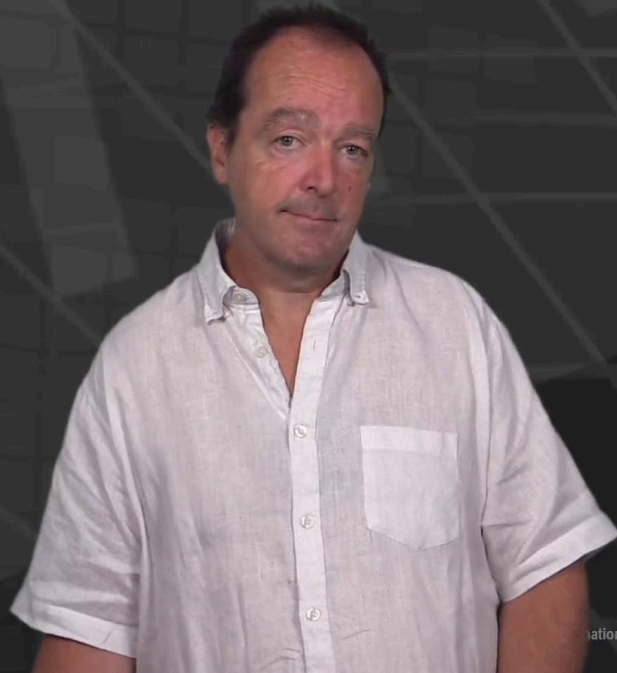
# Page Web – html et css

## Organisation des éléments de style

- Référencement des objets (type HTML, attributs «id» ou «class»), et
- Regroupement des éléments de style
- ➔ Dans une section de la page HTML
- ➔ Dans un fichier .css séparé

## Hiérarchie d'application des styles

- Défaut du navigateur >> feuille CSS >> section >> attribut
- Puis type >> classe >> objet



ation géographique

On voit donc que pour rationaliser la gestion des styles dans une page ou un ensemble de pages web, on va commencer par référencer les différents éléments HTML des pages web par leurs identifiants, leurs classes ou leurs types et que l'on va regrouper les informations de style, soit dans une section particulière caractérisée par la balise style dans l'en-tête de la page HTML, soit dans un fichier séparé de type.css. Avec cette façon de procéder, le style d'un élément peut être défini en fait à plusieurs endroits et s'applique alors la hiérarchie suivante où on passe d'abord du style par défaut de la page web au style défini dans la page CSS, puis dans la section de la page HTML puis comme attribut de l'objet lui-même. Et pareillement dans la hiérarchie de la typologie, on va aller d'abord du type général et puis si un style particulier est défini pour une classe particulière, c'est la classe qui va prendre le dessus et dans le cas où l'objet même a un style défini, c'est le style de l'objet qui va s'appliquer.

Notes

Summary



21m 55s



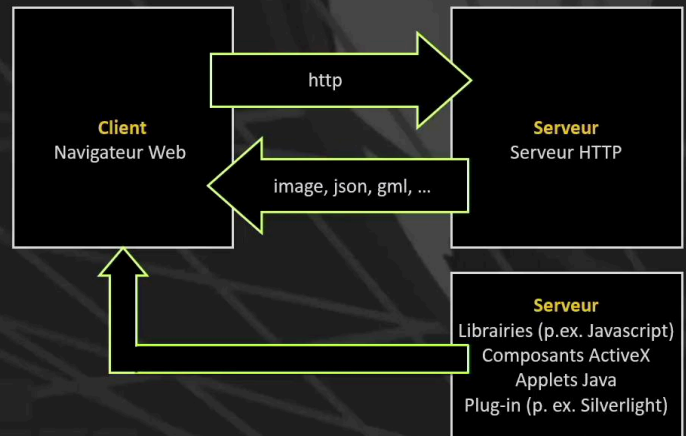
# Web SIG

HTML est limité

Ne suffit pas pour des applications complexes

Mais peut être complété, du côté client

- Par du code exécutable inséré dans la page HTML  
**javascript, vbscript, dhtml**
- Par du code exécutable à télécharger et utiliser avec une page HTML  
**scripts, activeX, java applet**
- Par des interfaces-applications spécifiques à installer sur le client – **plug-in**



Introduction aux systèmes d'information géographique

Nous avons donc vu les éléments de base de la construction et de la stylisation des pages web. Il nous reste encore à voir comment on va pouvoir intégrer dans ces pages web des éléments de cartographie et faire en sorte que cette cartographie puisse être interactive. Nous avons vu que le langage HTML est quand même limité et ne suffira pas pour ce type d'applications un peu plus complexes. Les possibilités qui s'offrent à nous, c'est de compléter en fait le HTML, ce qui peut se faire soit du côté client en ajoutant du code exécutable à l'intérieur même de la page HTML, ce code peut prendre la forme de Javascript, de vbscript ou du dhtml, par du code exécutable à télécharger et à utiliser avec une page HTML, il s'agit alors de scripts dans différents langages de programmation, du Python par exemple, des composants activeX ou des applets Java. Il existe une grande variété de produits disponibles pour ce genre de fonctions et par des interfaces applications quasiment complets, spécifiques, que l'on doit installer sur le client, ce que l'on désigne généralement sous le nom de plug-in.

Notes

Summary



23m 21s

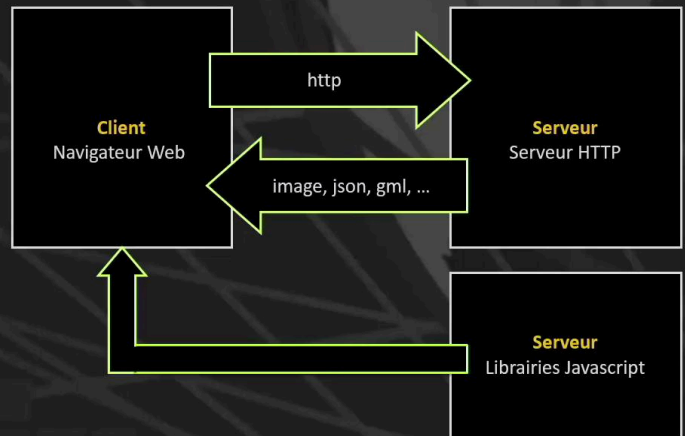
# Web SIG

HTML est limité

Ne suffit pas pour des applications complexes

Mais peut être complété, du côté client

- Par du code exécutable inséré dans la page HTML  
**javascript, vbscript, dhtml**
- Par du code exécutable à télécharger et utiliser avec une page HTML  
**scripts, activeX, java applet**
- Par des interfaces-applications spécifiques à installer sur le client – **plug-in**



Introduction aux systèmes d'information géographique

Les possibilités peuvent également être enrichies du côté serveur en ajoutant des exécutables ou des librairies DLL, voire même des scripts qui seraient exécutés sur le serveur et dont la fonction est en fait de créer de manière dynamique les pages HTML qui sont renvoyées au client. Et là, on utilise beaucoup le common gateway interface, le CGI, ou alors par du code exécutable qui est intégré à la page HTML mais qui est exécuté au niveau du serveur. L'exemple le plus fréquemment utilisé est celui du PHP, donc le code PHP qui se trouve dans la page en l'occurrence une page PHP envoyée par le client au serveur. Le serveur doit disposer d'un interpréteur PHP pour exécuter ce code, fabriquer la page HTML et la renvoyer au client. Alors, nous n'avons pas le temps de passer en revue de manière exhaustive toutes ces différentes formes d'enrichissement du HTML et nous allons nous concentrer dans la suite du cours sur la solution Javascript, donc du code exécutable Javascript inséré dans la page HTML et exécuté du côté client.

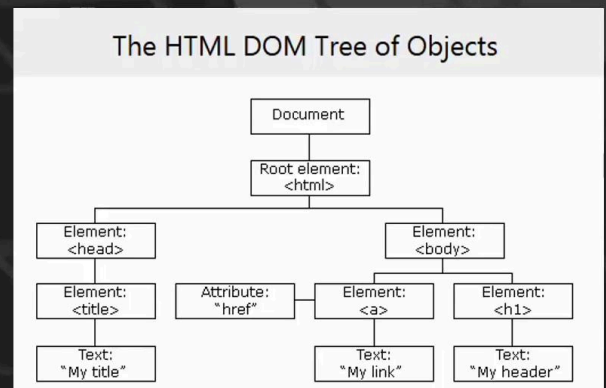
Notes

Summary



## Le Javascript

- Langage fonctionnel, i.e. que les fonctions sont des objets
- Les éléments de la page HTML sont accessibles via le **Document Object Model (DOM)**



Introduction aux systèmes d'information géographique

Le Javascript a, entre autres particularités, le fait d'être un langage fonctionnel, c'est-à-dire que les fonctions sont des objets. Ici, on a l'exemple d'une fonction de deux paramètres a et b qui renvoient le produit de ces deux variables. Cette fonction peut être définie dans une variable elle-même qui s'appellerait myFunction et on peut utiliser cette variable pour exécuter la fonction donc, dans le calcul de quatre fois trois. Le Javascript permet d'accéder aux éléments de la page HTML via le Document Object Model, ou DOM, qui comme on le voit sur cette illustration permet d'accéder alors à la page HTML qui est l'élément racine et puis à ses composants, l'en-tête et le corps et dans l'en-tête, le titre, le texte du titre ou dans le corps, les différents éléments HTML qui le composent.

Notes

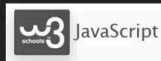
Summary



# Web SIG

## Le Javascript

- Langage fonctionnel, i.e. que les fonctions sont des objets
- Les éléments de la page HTML sont accessibles via le **Document Object Model (DOM)**
- Et les éléments du navigateur via le **Browser Object Model (BOM)**
- Plus de détails sur <http://www.w3schools.com/js>



## JS Browser BOM

JS Window  
JS Screen  
JS Location  
JS History  
JS Navigator  
JS Popup Alert  
JS Timing  
JS Cookies

Introduction aux systèmes d'information géographique

Il est également possible d'accéder aux éléments du navigateur via le BOM, Browser Object Model, qui permet d'accéder en fait à la fenêtre, à l'historique du navigateur, au type de navigateur, etc. A nouveau, si vous voulez approfondir vos connaissances sur le Javascript, je vous recommande de commencer par consulter le site w3schools qui propose toutes sortes d'exercices dans ce domaine.

Notes

Summary



27m 04s

# Web SIG

## HTML et Javascript

- Dans un fichier externe référencé par la page

myFile.html

```
<html>
<head>
  <script src="myScript.js"></script>
</head>
<body>
  <h4>This is a JavaScript example</h4>
  <button type="button"
    onclick="myFunction()">
    The time is?
  </button>
  <p id="demo"></p>
</body>
</html>
```

myScript.js

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Changed.";
}
```

Introduction aux systèmes d'information géographique

Nous allons nous intéresser maintenant à la manière d'intégrer du Javascript dans une page HTML avec pour commencer un exemple très simple d'un bouton dans une page HTML et on aimerait que lorsque l'on clique sur ce bouton, s'affichent le jour, la date, l'heure, etc. La syntaxe du fichier HTML est simple, donc, ici, pas d'en-tête, juste le corps, avec dans le corps, le titre, le bouton et puis un paragraphe avec l'ID demo qui doit recevoir en fait le texte de la date et de l'heure généré par la fonction. Et on voit que dans les attributs du bouton, on a une propriété onclick, donc un événement qui va exécuter la fonction qui est décrite entre guillemets. Donc, on va partir de document qui est donc le DOM, on va rechercher l'objet qui s'appelle demo, dont l'ID est demo, et puis, on va assigner au contenu HTML de cet objet la fonction Javascript date qui renvoie à la date courante. Tout comme pour les éléments de style, nous avons aussi la possibilité ensuite d'extraire l'ensemble du code Javascript d'une page HTML ou d'une série de pages HTML pour les regrouper dans une page js, Javascript, et puis de référencer ensuite ces différentes fonctions dans l'en-tête avec une balise script qui pointe sur le fichier Javascript tout en gardant au niveau de l'événement, onclick sur le bouton, l'événement myFunction qui est défini dans la page myScript.js.

Notes

Summary

27m 36s





# Web SIG

## HTML et Javascript

- Dans un fichier externe référencé par la page

myFile.html

```
<html>
<head>
  <script src="myScript.js"></script>
</head>
<body>
  <h4>This is a JavaScript example</h4>
  <button type="button"
    onclick="myFunction()">
    The time is?
  </button>
  <p id="demo"></p>
</body>
</html>
```

myScript.js

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Changed.";
}
```

Introduction aux systèmes d'information géographique

Ici à nouveau, ce procédé permet de simplifier et de faciliter la lecture de la page HTML et de regrouper tous les éléments Javascript à un même endroit.

Notes

Summary



29m 38s

# Web SIG

## Librairies SIG Javascript open source

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



Introduction aux systèmes d'information géographique

L'intégration d'une cartographie interactive dans une page web en utilisant du Javascript va donc passer par l'utilisation de librairies SIG Javascript.

Notes

Summary

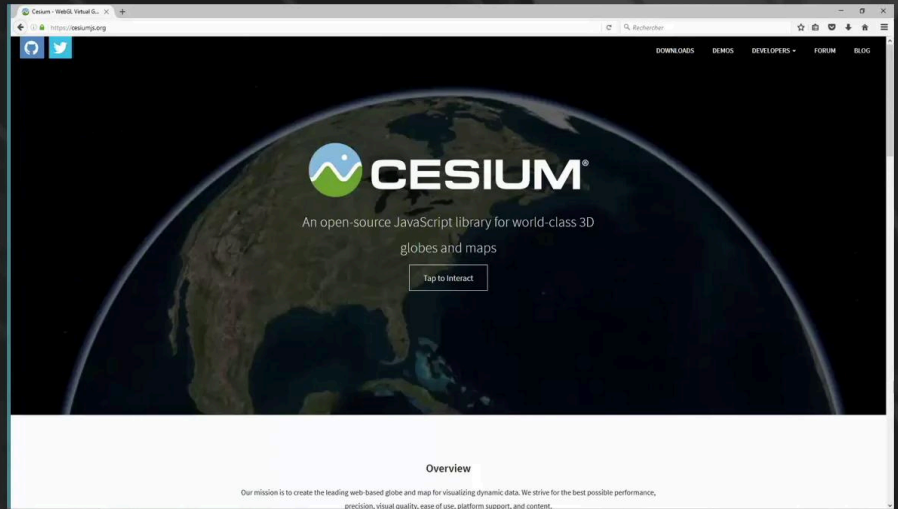


29m 51s

# Web SIG

## Librairies SIG Javascript open source

- 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
- ...



Introduction aux systèmes d'information géographique

Il existe un grand nombre de ces librairies parmi lesquelles en particulier Openlayers, Leaflet, la librairie Mapstraction, Polymaps, la librairie Cesium pour des représentations plutôt tridimensionnelles, etc.

Notes

Summary

30m 10s



# Web SIG

## Librairies SIG Javascript open source

- 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
- ...

## Autres librairies SIG Javascript

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

Introduction aux systèmes d'information géographique

Au delà de ces librairies open source, on trouve aussi une série de librairies qui sont en général libres, fournies par les grands producteurs que sont Google, Microsoft, ESRI, etc. C'est des librairies qui sont libres mais qui ne sont pas open source.

Notes

Summary



30m 28s

# Exemple d'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 <<<<title>Openlayer Seychelles</title>
6 </head>
7 <body>
8 </body>
```

Ajout de références

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**

- Création d'une page HTML de base
- Sauvegardée sous le nom **Openlayers\_Seychelles.html**
- Ajout du titre de la page
- Ajout d'une référence à la librairie Openlayers
- Ajout d'une référence à une feuille de styles par défaut

Introduction aux systèmes d'information géographique

Dans cette dernière partie du cours, nous allons démontrer comment utiliser une de ces librairies JavaScript cartographiques, en l'occurrence la librairie Openlayers, pour fabriquer une page HTML qui contient une carte géographique interactive. On commence donc par créer un fichier intitulé Openlayers\_Seychelles.html dans lequel on va retrouver les éléments de base, donc l'en-tête et le corps, avec un élément que l'on a encore pas vu qui est la déclaration de type !DOCTYPE html qui en fait se substitue aux balises html. On ajoute ensuite dans l'en-tête un titre, puis toujours dans l'en-tête, deux références, une première référence à la librairie Openlayers, et un lien vers un fichier CSS, un fichier de style par défaut qui accompagne la librairie Openlayers.

Notes

Summary

30m 54s





# Exemple d'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>
```

Eléments de style de la carte

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**

- Définition d'éléments de style pour la page
- Définition d'éléments de style pour la carte

Introduction aux systèmes d'information géographique

On ajoute ensuite dans le corps de la page un conteneur auquel on assigne l'IDMap et on définit quelques éléments de style dans une section style de l'en-tête avec tout d'abord pour l'ensemble de la page HTML et l'ensemble du corps, donc du body, la volonté d'occuper tout l'espace disponible, donc 100% de la hauteur et 100% de la largeur, et pour l'élément MAP, également 100 % de la hauteur et puis on donne une couleur de fond en gris clair pour pouvoir s'assurer que cet élément HTML carte, ce conteneur, va bien occuper tout l'espace disponible, ce que l'on peut vérifier en ouvrant le fichier dans un navigateur, en l'occurrence ici, Google Chrome.

Notes

Summary

32m 04s



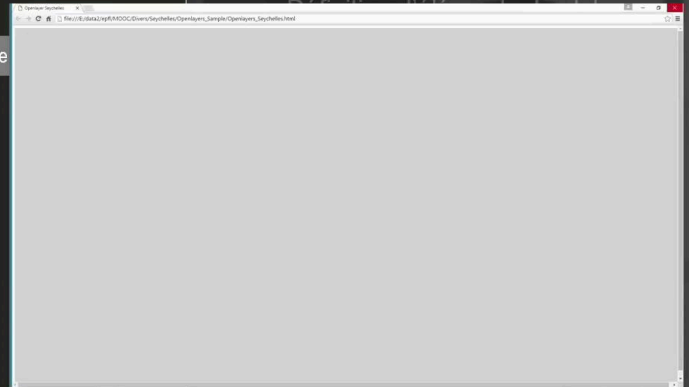
# Exemple d'application

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>
```

Eléments de

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**



Introduction aux systèmes d'information géographique

Et on voit qu'effectivement, le composant, le conteneur MAP, occupe bien tout l'espace disponible, à l'exception d'une petite marge en haut et à gauche.

Notes

Summary

33m 00s



# Exemple d'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
21   <script>
22     var map = new ol.Map({
23       target: 'map',
24       view: new ol.View({
25         center: ol.proj.fromLonLat([55.47, -4.67]),
26         zoom: 12
27       })
28     });
29
30     // create and add the tile layer
31     var osmlayer = new ol.layer.Tile({
32       source: new ol.source.OSM()
33     });
34     map.addLayer(osmlayer);
35
36   </script>
37 </body>
```

← Définition de la couche OpenStreetMap

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**

- Définition d'éléments de style pour la page
- Définition d'éléments de style pour la carte
- Ajout d'une section de code
- Ajout de la carte
- Ajout de la couche OpenStreetMap

Introduction aux systèmes d'information géographique

On ajoute ensuite une section pour héberger le code JavaScript avec les balises script et dans cette section script, on va ajouter le code qui permet de créer l'objet carte, donc on crée un nouvel objet OL pour openlayers.map donc un objet MAP de la librairie Openlayers que l'on assigne à la variable map et cet objet, dans ses propriétés, on dit que la cible que l'on vise, c'est le conteneur qui s'appelle map, donc on va mettre cet élément cartographique dans le conteneur, dans le div qui est l'IDMap et puis on définit une vue pour cette carte, donc on crée une nouvelle vue qui est centrée sur le point latitude, longitude 55 47 et -4 67 qui correspond à peu près au centre de l'île de Mahé et cette vue, on lui définit également, on peut lui définir comme attribut un niveau de zoom, ici la valeur 12. Pour que quelque chose soit affiché dans cette carte on va donc ajouter une couche, en l'occurrence la couche OpenStreetMap. Donc on crée une nouvelle couche de type tuile avec comme source OSM, qui est la source OpenStreetMap et on ajoute cette couche OSM layer à l'objet map qui est donc la carte Openlayers.

Notes

Summary

33m 13s



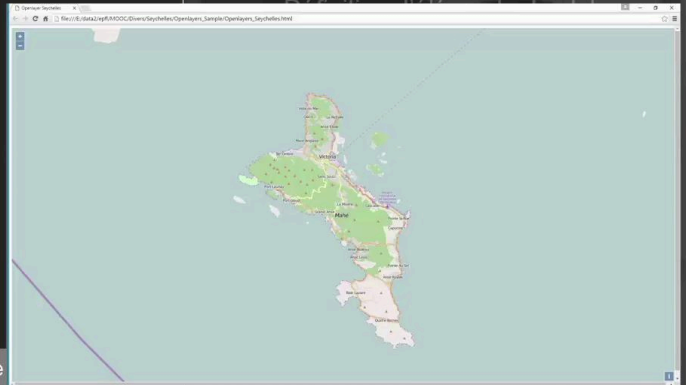
# Exemple d'application

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
21   <script>
22     var map = new ol.Map({
23       target: 'map',
24       view: new ol.View({
25         center: ol.proj.fromLonLat([55.47, -4.67]),
26         zoom: 12
27       })
28     });
29
30     // create and add the tile layer
31     var osmlayer = new ol.layer.Tile({
32       source: new ol.source.OSM()
33     });
34     map.addLayer(osmlayer);
35
36   </script>
37 </body>
```

← Définition de la couche

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**



Introduction aux systèmes d'information géographique

Et on peut constater que si l'on ouvre maintenant ce fichier HTML dans un navigateur comme Google Chrome, on a la carte OpenStreetMap des Seychelles, de l'île de Mahé, qui s'affiche dans la page web.

Notes

Summary



34m 47s

# Exemple d'application

Openlayers\_Seychelles.js

```
1  var map;  
2  
3  //Script executé lorsque la page est chargée  
4  
5  $(document).ready(function(){  
6  
7  
8      map = new ol.Map({  
9          target: 'map',  
10         view: new ol.View({  
11             center: ol.proj.fromLonLat([55.47, -4.67]),  
12             zoom: 12  
13         })  
14     });  
15  
16     // create and add the tile layer  
17     var osmlayer = new ol.layer.Tile({  
18         source: new ol.source.OSM()  
19     })  
20     map.addLayer(osmlayer);  
21  
22 });
```

Code exécuté au chargement de la page

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**

- Reprise du code javascript dans un fichier dédié
- Création de la carte et ajout de la couche OpenStreetMap au chargement de la page

Introduction aux systèmes d'information géographique

Pour ajouter du contenu et de l'interactivité à cette carte, il va falloir maintenant développer un petit peu le code JavaScript, raison pour laquelle on va sortir en fait ce script de la page web pour l'héberger dans un fichier JavaScript que l'on va intituler Openlayers\_Seychelles.js et on va référencer ce fichier dans l'en-tête de la page. En reprenant ce code JavaScript, on va commencer par séparer la définition de la variable map de l'utilisation de cette variable pour lui assigner la nouvelle carte Openlayers que l'on a créée, ce qui permet de mettre le code de création de la carte et d'ajout de la couche OpenStreetMap dans une fonction qui sera exécutée une seule fois au moment du chargement de la page, une fois que la page est chargée on exécute le code qui ajoute la carte, qui ajoute la couche OpenStreetMap avec les mots-clés "document ready", etc. A noter au passage que le symbole \$ se réfère à une librairie de JavaScript qui s'appelle jQuery, librairie qu'il a fallu pour pouvoir utiliser valablement ce symbole, librairie qu'il a fallu ajouter dans la page HTML, dans les références.

Notes

Summary

35m 03s





# Exemple d'application

Openlayers\_Seychelles.js

```
1
2 var map;
3
4 //Définition de quelques styles pour les couches vecteur
5 var District_Style = new ol.style.Style({
6   fill: new ol.style.Fill({ color: 'rgba(200, 10, 10, 0.2)' }),
7   stroke: new ol.style.Stroke({ color: 'rgba(255, 0, 0, 1)', width: 1 })
8 });
9
10 //Script executé lorsque la page est chargée
11
12 $(document).ready(function(){
13
14   map = new ol.Map({
15     target: 'map',
16     view: new ol.View({
17       center: ol.proj.fromLonLat([55.47, -4.67]),
18       zoom: 12
19     })
20   });
21
22   // create and add the tile layer
23   var osmlayer = new ol.layer.Tile({
24     source: new ol.source.OSM()
25   })
26   map.addLayer(osmlayer);
27
28   //Ajout de fichiers geojson
29   var vector = new ol.layer.Vector({
30     style: District_Style,
31     source: new ol.source.Vector({
32       url: './geojson/districts.geojson',
33       format: new ol.format.GeoJSON(),
34     })
35   });
36   map.addLayer(vector);
37
38 });
```

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**

- Reprise du code javascript dans un fichier dédié
- Création de la carte et ajout de la couche OpenStreetMap au chargement de la page
- Ajout de la couche geojson des districts
- Définition de la symbologie de la couche des districts

Introduction aux systèmes d'information géographique

Supposons maintenant que l'on veuille enrichir un petit peu notre carte, en ajoutant par exemple les districts des Seychelles que l'on a stocké ici dans un fichier JSON qui se trouve, on le voit par l'URL, dans un dossier GeoJSON qui se trouve lui-même au même niveau que notre fichier JavaScript et notre page HTML. Donc on crée une couche vecteur, avec un style qu'il va falloir définir, et on définit comme source de cette couche vecteur ce fichier GeoJSON, et on ajoute la couche à la carte. On définit donc ici la symbologie pour la couche des districts en créant un nouveau style composé d'un remplissage, donc un objet remplissage créé à partir d'une couleur et d'une transparence, et un objet stroke qui définit le style de trait pour la bordure des polygones basé également sur une couleur, une transparence, et sur une largeur. Alors on peut faire ensuite la même chose pour ajouter la couche des routes, la couche des hôtels et puis si on veut vraiment ajouter de l'interactivité, on pourrait écrire le code, on ne va pas le faire ici parce que ça devient un peu compliqué mais on pourrait écrire le code qui permet, lorsque l'on clique sur un district par exemple ou sur un hôtel d'avoir une petite fenêtre pop-up qui s'ouvre dans la page web et qui donne la liste des propriétés, des attributs de l'hôtel, du district, etc.

Notes

Summary

36m 29s



# Exemple d'application

Pour que cela fonctionne...

- Il faut encore rendre le fichier `districts.json` accessible sur un serveur
- Un simple serveur statique qui se contente de servir des fichiers
- Par exemple avec Node.js, dans un fichier nommé `server.js`  
<https://nodejs.org>

Introduction aux systèmes d'information géographique

Mais pour que tout cela fonctionne, il faut encore rendre ces fichiers JSON accessibles, donc il faut pouvoir les exposer sur un serveur, en l'occurrence on peut se contenter d'un simple serveur statique qui ne va pas faire de manipulations et qui va simplement servir ces fichiers. Serveur statique que l'on peut créer avec la librairie `node.js`, donc également du JavaScript, dans un fichier qu'on va appeler `server.js` et on voit qu'en fait, en cinq lignes de code, on crée ce serveur.

Notes

Summary

38m 08s



# Exemple d'application

server.js

```
1 var express = require('express');
2 var app = express();
3
4 app.use(express.static(__dirname));
5 app.listen(process.env.PORT || 3000);
6 console.log('server up and running. Listening on port 3000')
```

Avec ces quelques lignes de code,  
on crée un serveur **Node Express** statique  
qui va écouter le trafic arrivant sur le port 3000

Pour que cela fonctionne...

- Il faut encore rendre le fichier `districts.json` accessible sur un serveur
- Un simple serveur statique qui se contente de servir des fichiers
- Par exemple avec Node.js, dans un fichier nommé `server.js` <https://nodejs.org>

Introduction aux systèmes d'information géographique

On utilise une variante de node qui s'appelle Node Express qui est une forme simplifiée de serveur Node que l'on définit en fait le dossier public dans lequel se trouvent les fichiers que l'on va rendre accessible, qui est en l'occurrence le dossier dans lequel se trouve ce fichier `serveur.js` et toute l'arborescence qui est située en aval. On va dire que le serveur doit écouter le port 3000 et puis envoyer un message à la console pour dire que le serveur a démarré et qu'il écoute le port 3000.

Notes

Summary

38m 45s



# Exemple d'application

Nom	Modifié le	Type	Taille
geojson	17.05.2016 14:33	Dossier de fichiers	
node_modules	17.05.2016 09:36	Dossier de fichiers	
Openlayers_Seychelles.html	17.05.2016 12:05	Fichier HTML	1 Ko
Openlayers_Seychelles.js	17.05.2016 12:04	Fichier de JavaScript	1 Ko
server.js	17.05.2016 14:32	Fichier de JavaScript	1 Ko

Nom	Modifié le	Type	Taille
districts.geojson	17.05.2016 14:33	Fichier GEOJSON	1 487 Ko
hotels.geojson	17.05.2016 14:33	Fichier GEOJSON	31 Ko
routes.geojson	17.05.2016 14:33	Fichier GEOJSON	2 600 Ko

Création d'une page Web SIG pour l'île de Mahé aux Seychelles, avec la librairie **Openlayers 3**

- Structure de fichiers dans le dossier **Openlayers\_Sample**
- Démarrage du serveur node

```
Invite de commandes - node server.js
Microsoft Windows [version 10.0.10586]
(c) 2015 Microsoft Corporation. Tous droits réservés.

C:\Users\marc>e:

E:\>cd data2\epfl\MOOC\Divers\Seychelles\Openlayers_Sample

E:\data2\epfl\MOOC\Divers\Seychelles\Openlayers_Sample>node server.js
server up and running. Listening on port 3000
```

Introduction aux systèmes d'information géographique

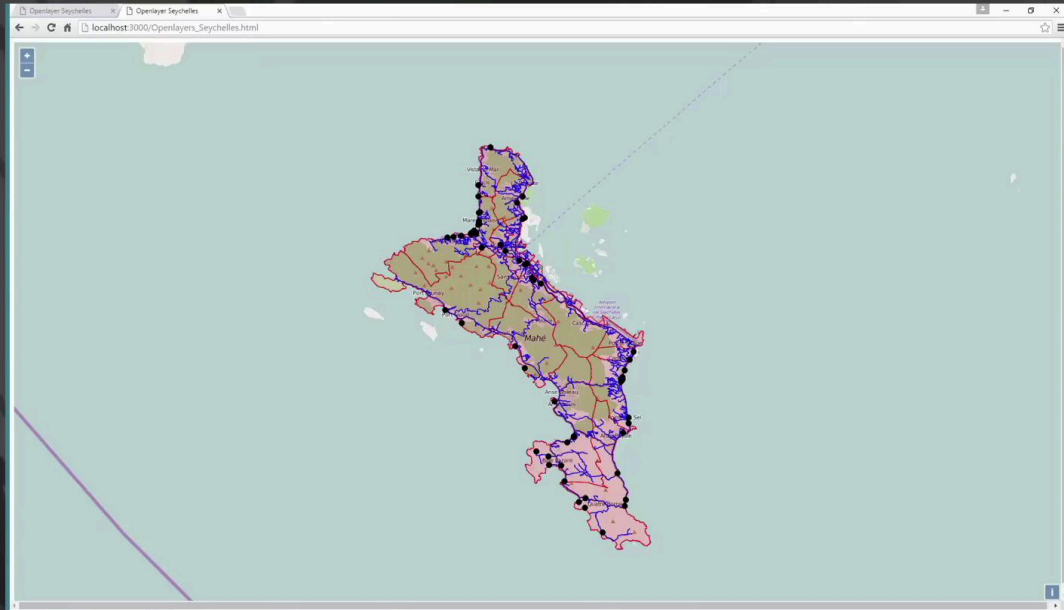
Donc si on jette un petit coup d'œil sur l'arborescence des fichiers que l'on a créé, on a donc dans le dossier Openlayers\_Samples ces trois fichiers HTML JavaScript pour la page web elle-même server.js pour le serveur, les différents modules de la librairie node qui vont permettre de faire tourner ce serveur, et dans un dossier GeoJSON, on va trouver les trois fichiers contenant les districts, les hôtels et les routes des Seychelles. On voit ici que l'étape suivante consiste, dans une fenêtre terminale, à descendre dans l'arborescence jusqu'à ce dossier Openlayers\_Samples et puis d'envoyer la commande, node serveur.js ce qui va lancer le serveur node et on a le message comme quoi ce serveur a démarré et qu'il écoute les requêtes sur le port 3000.

Notes

Summary



# Exemple d'application



Introduction aux systèmes d'information géographique

Pour charger notre page web, il faut donc ouvrir un navigateur Internet et comme url taper localhost puisqu'on est sur une machine locale, :3000 pour rechercher le port 3000, donc le serveur node que l'on vient de démarrer et puis accéder à la page Openlayers\_Seychelles.html qui va exécuter le code JavaScript associé, qui va lui-même aller chercher les fichiers GeoJSON et qui va les ajouter sur la carte, ce qui fait que l'on retrouve ici en fait les districts, les routes et les hôtels. Et comme dit tout à l'heure, on peut développer ce code pour ajouter de l'interactivité à cette carte sous toute sorte de forme.

Notes

Summary



40m 29s