

Cartographie Geolocalisation et capteurs

Christophe Vestri

Le mercredi 14 janvier 2022

Plan du cours

- 7 janvier : Intro, github, Capteur/Geoloc en HTML5
- 14 janvier: carto/geo, leaflet/mapBox, rest Api
- 25 janvier: 2D/3D: Canvas, WebGL et Three.js
- 1 février: Aframe/AR.js, exercice + projet
- 8 février : Projets

Objectifs du cours

- Bases de geolocalisation et de la cartographie
- Expérimenter quelques méthodes et outils web geo/3D
- Réaliser un projet

<https://github.com/vestri/CoursGeo>

- Evaluation:
 - Exos des cours (50%)
 - Projet (50%)

Plan Cours 2

- Debugging
- Référentiels
- Exercices

- TD1
- Repères Géographiques et Cartographiques
- Exercices en Html5/javascript
 - Leaflet, openStreetmap
 - MapBox, mapQuest
 - REST API

Outils de debug

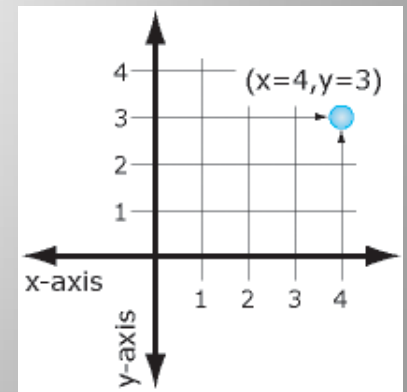
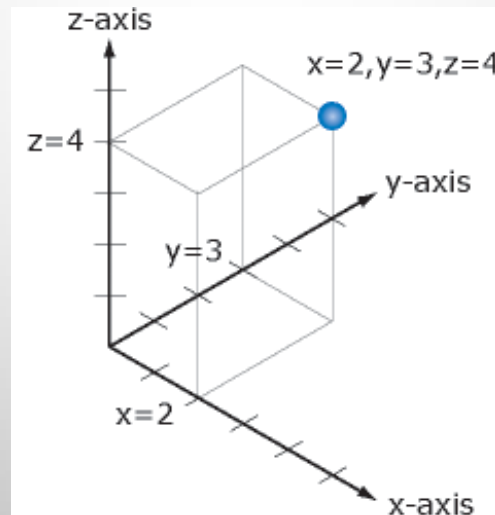
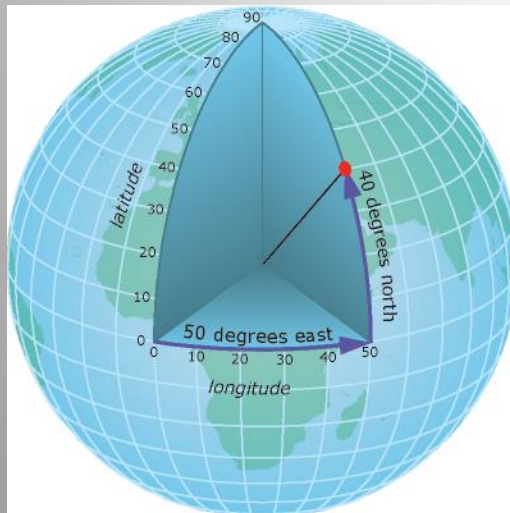
- En local:
 - `python3 -m http.server`
 - <http://localhost:8000/> firefox ou chrome
- Smartphone android -> Chrome
- <https://developers.google.com/web/tools/chrome-devtools/javascript>
 - Simulation de smartphone (F12)
 - Connecté à un smartphone: <chrome://inspect/>
- Firefox possible ou autres??

Les Systèmes de Coordonnées de Référence

- Debugging
- **Référentiels**
- Exercices

Les SCR (Systèmes de Coordonnées de Référence) sont des modèles mathématiques permettant, grâce aux coordonnées, de faire le lien entre un endroit réel sur terre et sa représentation en plan.

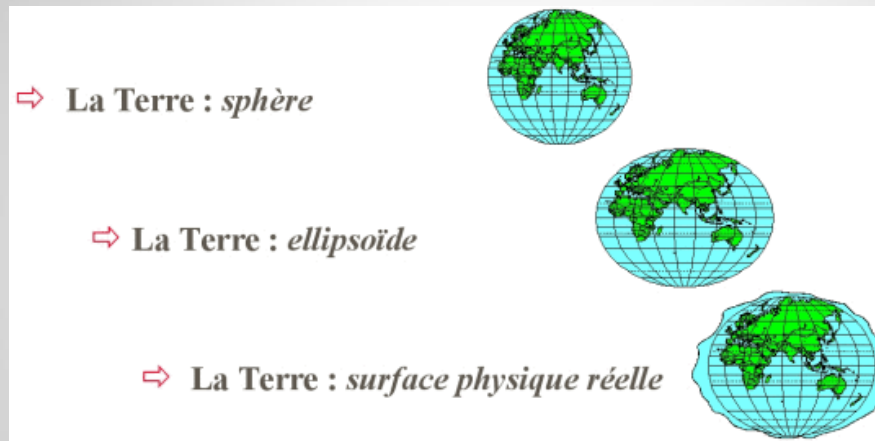
En général, les SCR se divisent en systèmes de coordonnées de référence projetées (aussi appelés systèmes de coordonnées de référence cartésiennes ou rectangulaires) et systèmes de coordonnées de référence géographique.



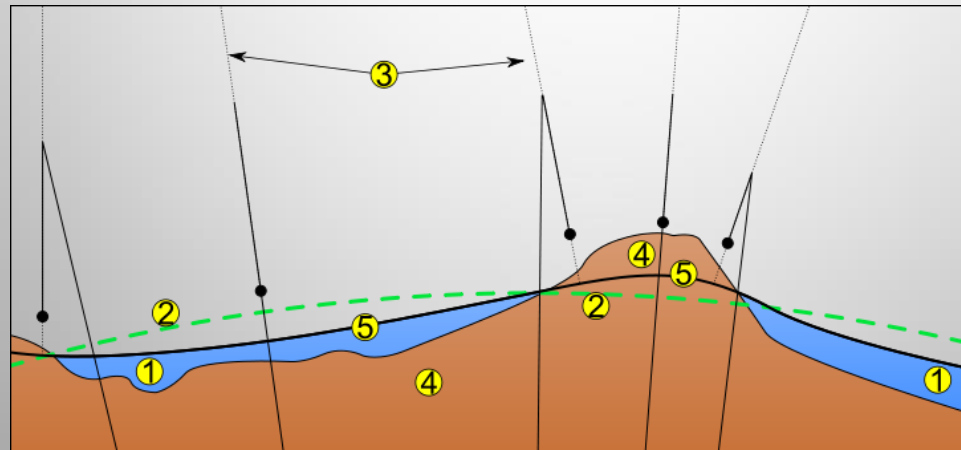
Systèmes Géographiques et Cartographiques

- Debugging
- **Référentiels**
- Exercices

- Représentation de la terre



Géoïde



— Ellipsoïde

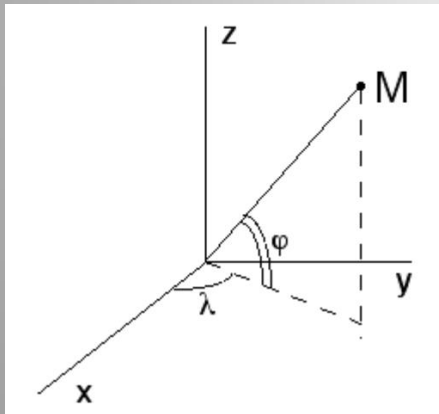
Systèmes Géographiques et Cartographiques

- Debugging
- **Référentiels**
- Exercices

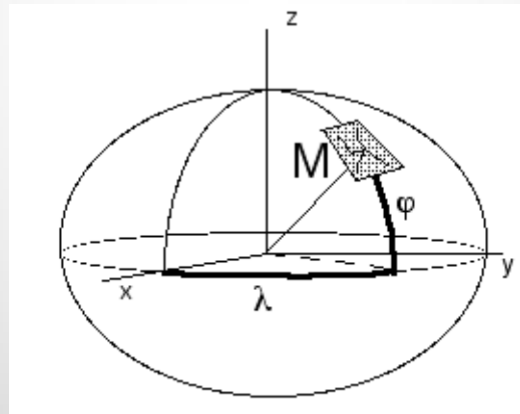
- Construction d'un référentiel géographique

Choix d'un ellipsoïde

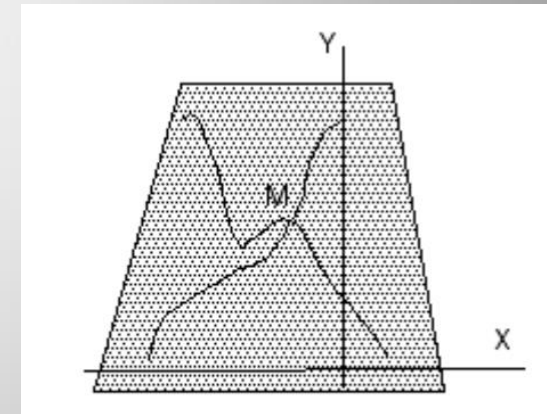
Choix d'une projection



Système cartésien
 x, y, z



Système géographique
 φ, λ



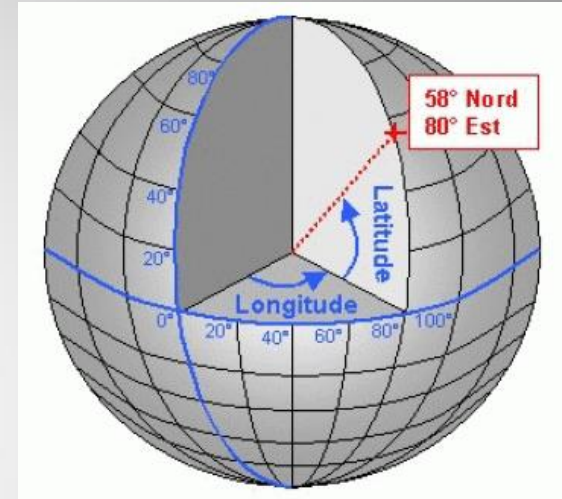
Système cartographique
 X, Y

Systèmes Géographiques et Cartographiques

- Géolocalisation
- Référentiels
- Capteurs
- Git
- Exos

- Un point de la surface terrestre est repéré en fonction d'un ellipsoïde par :

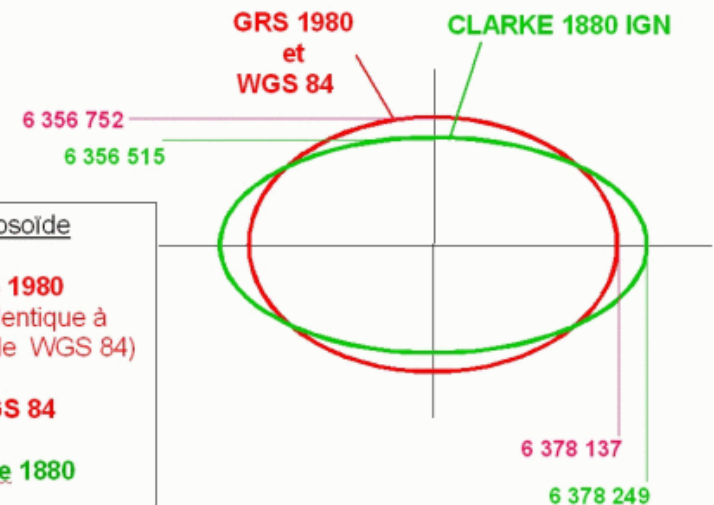
- sa longitude : λ (Lambda)
- sa latitude : ϕ (Phi)



- Différents systèmes:

- GPS ([WGS84](#)),
- Europe ([ETRS89](#))
- France ([NTF](#), [RGF 93](#))

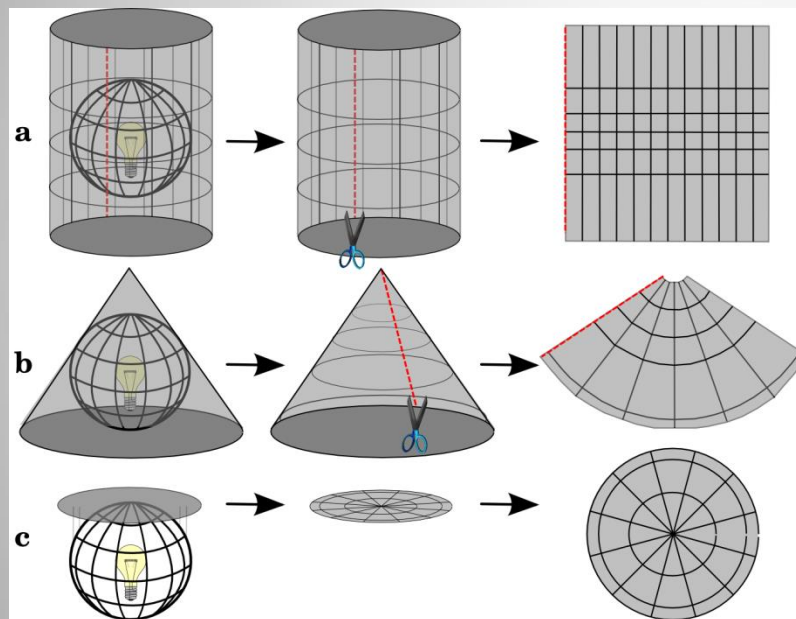
Systèmes	Ellipsoïde
RGF93	GRS 1980 (quasi identique à l'ellipsoïde WGS 84)
WGS84	WGS 84
NTF	Clarke 1880



Systèmes Géographiques et Cartographiques

- Debugging
- **Référentiels**
- Exercices

Les 3 familles de projections cartographiques :



- a) Projections cylindriques
- b) Projections coniques
- c) Projections planes ou azimutales

Chaque [projection cartographique](#) a des avantages et des désavantages. La meilleure projection d'une carte dépend de l'échelle de la carte, et pour l'objectif pour laquelle elle sera utilisée.

Les projections et SCR

- Debugging
- **Référentiels**
- Exercices

Les projections

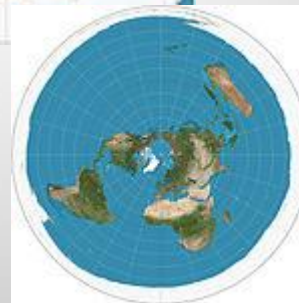
Distorsions des projections cartographiques

Equidistante : conserve les distances

Equivalente : conserve les surfaces => intérêt : petite échelle

Conforme ou **orthomorphique** : conserve les formes et les angles localement

Aphylactique : ne conserve ni angles, ni surfaces



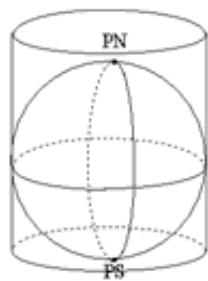
Systemes Géographiques et Cartographiques

- Debugging
- **Référentiels**
- Exercices

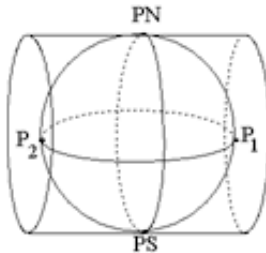
<https://docs.mapbox.com/mapbox-gl-js/example/>

- Choix d'une projection cartographique

Représentation cylindrique :

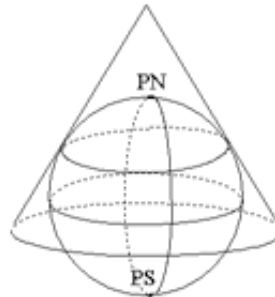


directe

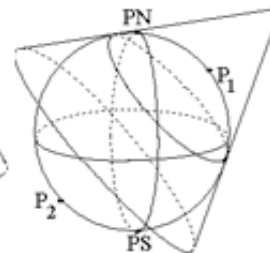


transverse

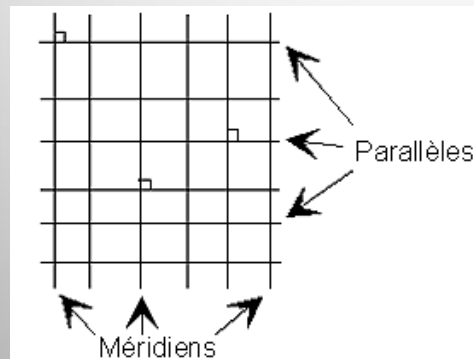
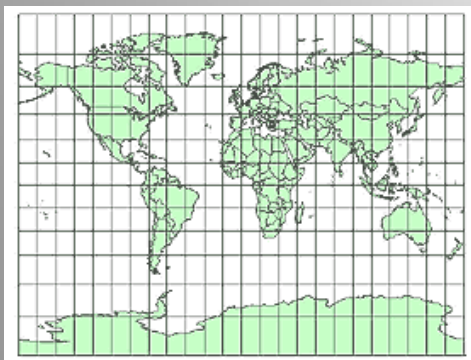
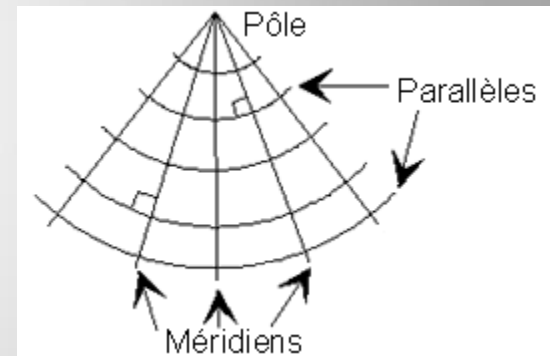
Représentation conique :



directe



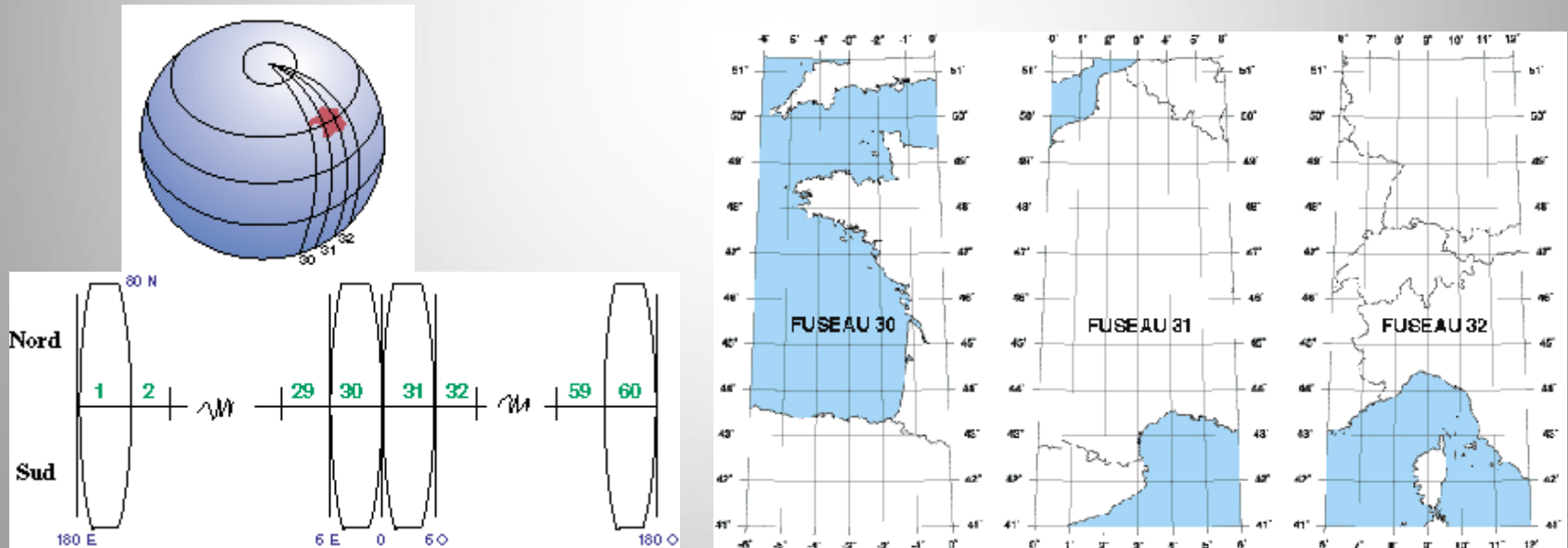
oblique



Systèmes Géographiques et Cartographiques

- Debugging
- Référentiels
- Exercices

- GPS: UTM (Universal Transverse Mercator)
 - Système mondial de 122 projections
 - 60 **fuseaux** de 6° (entre 80° Sud et 80° Nord) + 2 poles

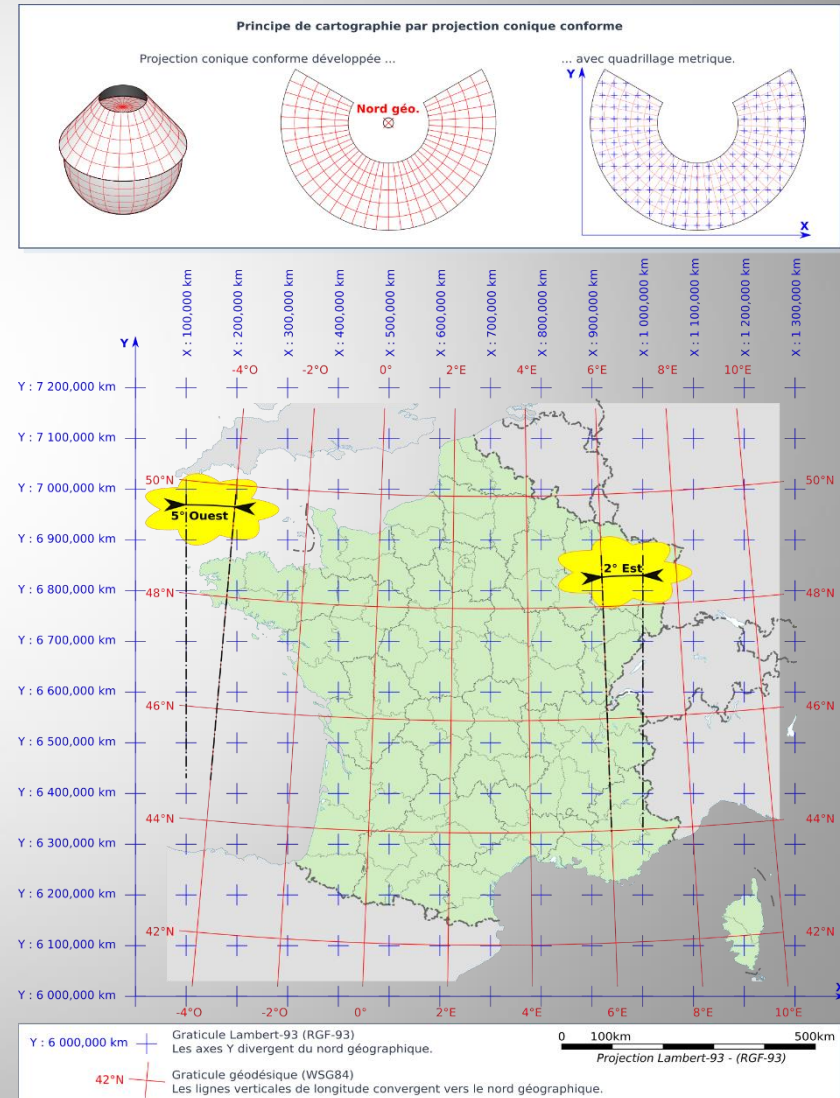


- La France: fuseaux UTM Nord 30, 31 et 32

- Debugging
- **Référentiels**
- Exercices

-
- A diagram of Earth showing its rotation axis. The axis is a vertical line passing through the center of the Earth, with arrows at both ends. The text "Axe de rotation de la Terre" is written above the top arrow. A vertical double-headed arrow indicates the radius, labeled "6356km". A horizontal double-headed arrow indicates the equatorial radius, labeled "6378km".

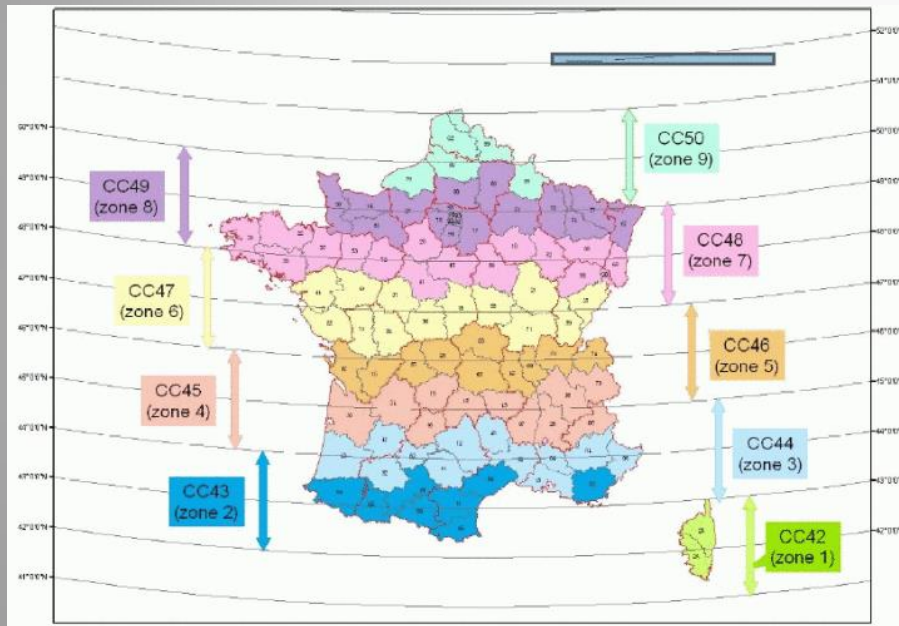
-



Systèmes géographique Français

- Debugging
- **Référentiels**
- Exercices

- Système géographique Français Lambert CC42...



Projection	φ_0	φ_1	φ_2	X_0	Y_0	EPSG
CC42	42°	41.25°	42.75°	1 700 000 m	1 200 000 m	3942
CC43	43°	42.25°	43.75°	1 700 000 m	2 200 000 m	3943
CC44	44°	43.25°	44.75°	1 700 000 m	3 200 000 m	3944
CC45	45°	44.25°	45.75°	1 700 000 m	4 200 000 m	3945
CC46	46°	45.25°	46.75°	1 700 000 m	5 200 000 m	3946
CC47	47°	46.25°	47.75°	1 700 000 m	6 200 000 m	3947
CC48	48°	47.25°	48.75°	1 700 000 m	7 200 000 m	3948
CC49	49°	48.25°	49.75°	1 700 000 m	8 200 000 m	3949
CC50	50°	49.25°	50.75°	1 700 000 m	9 200 000 m	3950

- 9 projections appelées coniques conformes 9 zones

Systèmes Géographiques et Cartographiques

- Debugging
- **Référentiels**
- Exercices

- Coordonnées GPS: Lat/Lon
 - La salle 202:
 $43.616513, 7.072094 = 43^{\circ}36'59.5''N + 7^{\circ}04'19.5''E$
- Plus d'infos:
 - Wikipédia
 - IGN: <http://geodesie.ign.fr/index.php> et <http://education.ign.fr/dossiers/mesurer-la-terre>
 - <http://seig.ensg.eu/>
 - http://sgcaf.free.fr/pages/techniques/ign_cooronnees.htm

Leafletjs

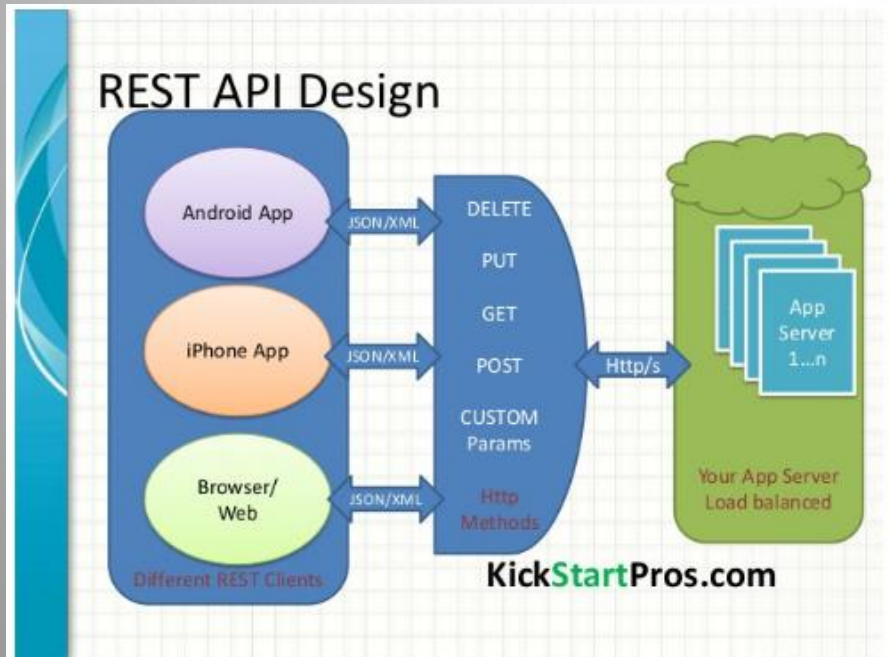
- Debugging
- Référentiels
- Exercices

- [leafletjs](https://leafletjs.com/) est une librairie Opensource pour afficher des cartes interactives utiles à la navigation (comme google maps)
- Seulement 33Ko, Tous les browsers
 - Map controls
 - Layers
 - Interaction Features
 - Custom maps



REST API

- REST (*representational state transfer*)
- Accès simple à des webservice
- <https://ensweb.users.info.unicaen.fr/pres/ws/>
- <https://www.uptrends.fr/qu-est-ce-que/rest-api>



Contraintes

- Client-serveur
- Sans état
- Avec/sans cache
- En couche
- Interface uniforme
- (code à la demande)

REST API

Exemple de hierarchie: <https://api.gouv.fr/api/api-geo.html>

<https://blog.octo.com/designer-une-api-rest/>

API	Domaines / Sous domaines	Exemples d'URI
Google	https://accounts.google.com https://www.googleapis.com https://developers.google.com	https://accounts.google.com/o/oauth2/auth https://www.googleapis.com/oauth2/v1/tokeninfo https://www.googleapis.com/calendar/v3/ https://www.googleapis.com/drive/v2 https://maps.googleapis.com/maps/api/js?v=3.exp https://www.googleapis.com/plus/v1/ https://www.googleapis.com/youtube/v3/ https://developers.google.com
Facebook	https://www.facebook.com https://graph.facebook.com https://developers.facebook.com	https://www.facebook.com/dialog/oauth https://graph.facebook.com/me https://graph.facebook.com/v2.0/{achievement-id} https://graph.facebook.com/v2.0/{comment-id} https://graph.facebook.com/act_{ad_account_id}/adgroups https://developers.facebook.com
Twitter	https://api.twitter.com https://stream.twitter.com https://dev.twitter.com	https://api.twitter.com/oauth/authorize https://api.twitter.com/1.1/statuses/show.json https://stream.twitter.com/1.1/statuses/sample.json https://dev.twitter.com
GitHub	https://github.com https://api.github.com https://developer.github.com	https://github.com/login/oauth/authorize https://api.github.com/repos/octocat/Hello-World/git/commits/7638417db6d59f3c431d3e1f261cc637155684cd https://developer.github.com

Exercices 1

- Avec Leafletjs
 - Récupérez votre position GPS, afficher votre position
 - Afficher une carte locale (utilisez openStreetmap)
 - Affichez un marqueur sur Nice

Testez en local puis publiez sur Github

Exercices 2

- Avec Leafletjs
 - Tracez le triangle des Bermudes (en rouge)
 - Changer de carte (stamen:
<http://maps.stamen.com/>)
 - Dessiner un cercle autour de votre position avec une rayon représentant la précision estimée
 - Calculez la distance à Marseille, l'afficher
(https://fr.wikipedia.org/wiki/Distance_du_grand_cercle)

Exercices 3

- Avec Leafletjs ou autre, récupérer des données géoréférencées et les afficher sur la carte
 - Geojson sur <http://opendata.nicecotedazur.org>
 - ou par une RestApi :
 - <https://www.insee.fr/fr/metadonnees/cog/departement/DEP06-alpes-maritimes>
 - <https://www.data.gouv.fr/fr/>
 - <https://api.gouv.fr/api/api-geo.html>
 - <https://adresse.data.gouv.fr/api>
 - Bonus:
 - afficher un trajet/route (google/mapbox/mapQuest)
 - Testez d'autres outils
 - [mapQuest](#) (Token: tR2C6osuQcc3RoWnxDMXF6FACtNAzMl8)
 - [mapBox](#), [google maps api](#)