



# **GEOREFERENZIAMENTO E GOOGLE EARTH TUTORIAL**

**Corso di Basi di Dati II Mod. B**

**Prof. F. Cutugno**

**Slide a cura di Leano Vincenza Anna**

**[leano@na.infn.it](mailto:leano@na.infn.it)**



# GEOREFERENZIAZIONE E RAPPRESENTAZIONE SPAZIALE

# GEOREFERENZIAZIONE

- Attribuire ad un dato informazioni sulla sua dislocazione geografica
- Definire l'esistenza di un oggetto in uno spazio fisico.
- Definire la sua posizione in termini di coordinate (latitudine e longitudine)
- Le coordinate solo relative a un sistema geodetico di riferimento.



# SISTEMA GEODETICO DI RIFERIMENTO

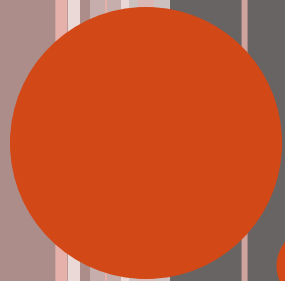
- Definisce il meridiano di origine e la forma dell'ellissoide con cui si approssima il globo terrestre.
- Effetto della scelta: lo stesso punto può essere indicato con coordinate (leggermente) diverse
- Sistemi esistenti:
  - Roma40: basato sull'ellissoide Internazionale di Hayford orientato sulla verticale di Roma - Monte Mario. Origine delle longitudini sul meridiano di Monte Mario.
  - ED50 (European Datum 1950): basato sull'ellissoide Internazionale di Hayford, orientamento medio europeo. Origine delle longitudini sul meridiano di Greenwich.
  - WGS84 (World Geodetic System 1984): basato su uno specifico ellissoide geocentrico con schiacciamento ai poli. Origine delle longitudini sul meridiano di Greenwich.



# RAPPRESENTAZIONE SPAZIALE

- Un oggetto georeferenziabile può essere rappresentato sulla mappa o sul geobrowser come:
  - Punto
  - Linea
  - Poligono





# GOOGLE EARTH

Breve Tutorial



# GOOGLE EARTH

- È un mappamondo virtuale.
- È liberamente disponibile per uso personale.
- Permette di esplorare in maniera tridimensionale località sulla terra e nello Spazio.
- Permette di vedere gli edifici in 3D
- Consente di visualizzare un'elevazione del terreno
- Etc...



## CARATTERISTICHE

- Mappe disponibili in streaming dai server di Google
- Utilizzo del sistema WGS84 (World Geodetic System) per le coordinate geografiche
- Disponibile come versione stand-alone per PC o come un plug-in per browser





# GOOGLE EARTH DESKTOP

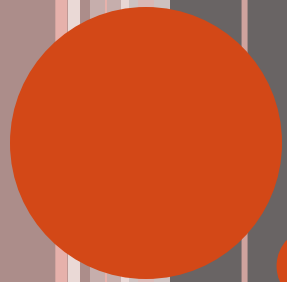
- La versione 5 è disponibile per i sistemi Operativi: Microsoft Windows (2000, XP, Vista, 7), Mac OS (X 10.3.9), Linux (dal 12 giugno 2006) e FreeBSD.
- Sorgenti di informazioni (*layer*) incluse:
  - Geographic Web
  - Roads
  - Label and Borders
  - Terrain
  - 3D Buildings
- Altre caratteristiche:
  - Street View
  - Immagini storiche.



# GOOGLE EARTH PLUGIN

- Permette di integrare il geobrowser nelle pagine web
- Disponibile per i seguenti sistemi operativi e browser:
  - **Microsoft Windows (2000, XP, Vista, 7):** Google Chrome dalla versione 1.0, Internet Explorer dalla versione 6.0, Firefox dalla versione 2.0 e Flock dalla versione 1.0.
  - **Apple Mac OS X 10.4 e versioni successive (Intel e PowerPC):** Safari dalla 3.1, Firefox dalla 3.0, Chrome dalla 1.0
- Ha delle limitazioni sui layer e sui tag KML riconosciuti.
  - Layer disponibili: Terrain, Borders and labels, Roads, 3D buildings.





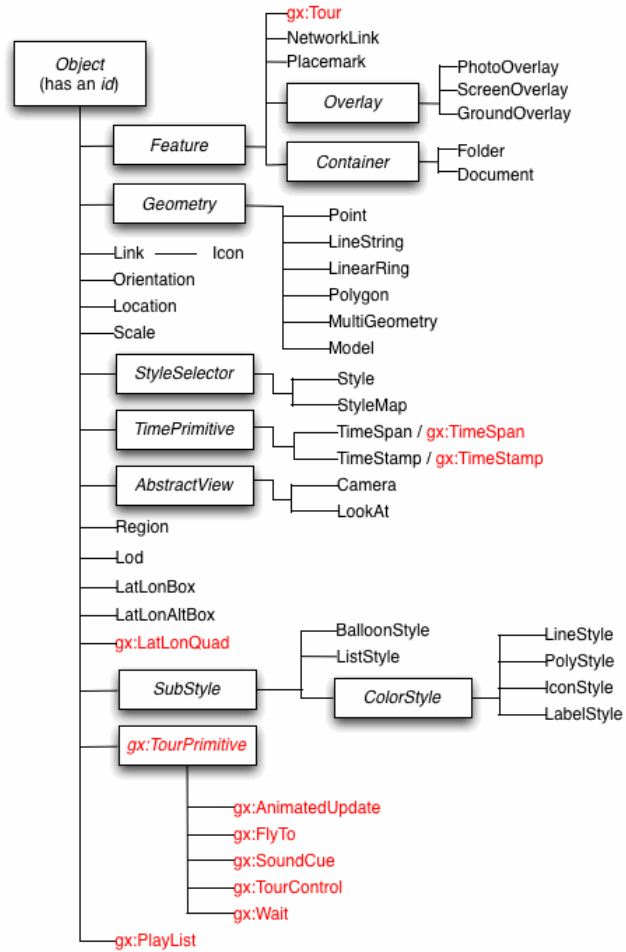
**KML**



## KML: KEYHOLE MARKUP LANGUAGE

- Linguaggio basato su una grammatica XML creato per visualizzare caratteristiche e oggetti geografici su mappe o geobrowser
- Ideato dalla Keyhole Inc., acquisita da Google nel 2004.
- KML 2.2 è stato dichiarato uno Standard dall'Open Geospatial OGC (Shankland, 2008).
- Coordinate geografiche tridimensionali: longitudine, latitudine e altitudine in quest'ordine.
- I file KML spesso sono distribuiti in file KMZ, che sono dei file compressi con estensione .kmz.





## STRUTTURA DI UN FILE KML

- XML header.
- KML namespace declaration. This is line 2 in every KML 2.2 file.
- Un oggetto Placemark che può contenere i seguenti elementi:
  - *name* usato per la label del placemark
  - *description* Che appare nel "balloon" del Placemark
  - 0 o più elementi *Geometry* (Point, LineString, Polygon...).
- N.B: Solo un placemark che avrà tra le geometrie un point avrà la classica iconcina gialla dei placemark di google e una label.



## ESEMPIO KML

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Naples</name>
    <description>Città di Napoli</description>
    <Point>
      <coordinates>14.151575, 40.502604, 0</coordinates>
    </Point>
  </Placemark>
</kml>
```



# VISUALIZZAZIONE SU GOOGLE EARTH





## CONTENITORI

- Se il file contiene più placemark, essi dovranno essere raccolti in:
  - <Document>: raccoglie i placemark e permette di definire stili comuni
  - <Folder>: permette di organizzare i placemark Gerarchicamente



## STILE

**<Style id="ID">**

*<!-- extends StyleSelector -->*

*<!-- specific to Style -->*

*<IconStyle>...</IconStyle>*

*<LabelStyle>...</LabelStyle>*

*<LineStyle>...</LineStyle>*

*<PolyStyle>...</PolyStyle>*

*<BalloonStyle>...</BalloonStyle>*

*<ListStyle>...</ListStyle>*

**</Style>**



## PLACEMARK

**<Placemark id="ID">**

*<!-- inherited from Feature element -->*

**<name>...</name>** *<!-- string -->*

**<visibility>1</visibility>** *<!-- boolean -->*

**<description>...</description>** *<!-- string -->*

*<!-- Camera or LookAt -->*

**<TimePrimitive>...</TimePrimitive>**

**<styleUrl>...</styleUrl>** *<!-- anyURI -->*

**<StyleSelector>...</StyleSelector>**

*<!-- specific to Placemark element -->*

**<Geometry>...</Geometry>**

**</Placemark>**



## GEOMETRY: POINT

**<Point id="ID">**

*<!-- specific to Point -->*

**<extrude>0</extrude>** *<!-- boolean -->*

**<altitudeMode>clampToGround</altitudeMode>**

*<!-- kml:altitudeModeEnum: clampToGround,  
relativeToGround, or absolute -->*

*<!-- or, substitute gx:altitudeMode: clampToSeaFloor,  
relativeToSeaFloor -->* **<coordinates>...</coordinates>**

*<!-- lon,lat[,alt] -->*

**</Point>**

- Coordinate: Coppia lon, lat (eventualmente tripla con alt)



## GEOMETRY: LINESTRING

- Rappresentano dei punti uniti da segmenti lineari.

**<LineString id="ID">**

<!-- specific to LineString -->

<extrude>0</extrude> <!-- boolean -->

<tessellate>0</tessellate> <!-- boolean -->

<altitudeMode>clampToGround</altitudeMode>

<coordinates>...</coordinates> <!-- lon,lat[,alt] -->

**</LineString>**

- Coordinate: Almeno due coppie/triple Lon, lat (,alt) separate da spazi



## GEOMETRY: POLYGON

- Un poligono è formato da un outer boundary e zero o più inner boundary.
- Ogni Boundary è caratterizzato da un elemento LinearRing che ne determina la geometria.
- Le coordinate sono un insieme di coppie (triple) in cui la prima e l'ultima coppia sono uguali (serve per chiudere il poligono)



# POLYGON

**<Polygon id="ID">**

*<!-- specific to Polygon -->*

**<extrude>0</extrude>** *<!-- boolean -->*  
    **<tessellate>0</tessellate>** *<!-- boolean -->*  
    **<altitudeMode></altitudeMode>**

**<outerBoundaryIs>**

**<LinearRing>**  
        **<coordinates>...</coordinates>**  
    **</LinearRing>**

**</outerBoundaryIs>**

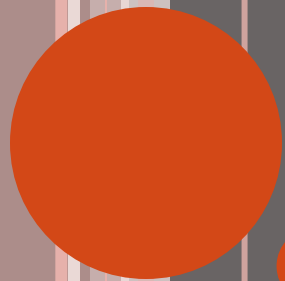
**<innerBoundaryIs>**

**<LinearRing>**  
        **<coordinates>...</coordinates>**  
    **</LinearRing>**

**</innerBoundaryIs>**

**</Polygon>**





**API GOOGLE**



# INSTALLAZIONE

- Scaricare il plugin da <http://code.google.com/apis/earth>
- Procurarsi una Api Key:
  - <http://code.google.com/intl/it-IT/apis/maps/signup.html>
  - Si necessita di una key diversa per ogni Url.



## INTEGRAZIONE IN UNA PAGINA WEB

- Google earth viene integrato in una pagina web utilizzando le google API in javascript.
- Per instanziare una vostra copia di google earth bisogna inserire nella sezione HEAD il tag:
  - `<script type="text/javascript" src="http://www.google.com/jsapi?key=ABCDEF"> </script>`
- Le invocazioni alle API google saranno fatte tramite
  - `<script type="text/javascript" > Some Code Here</script>`
  - `<script type="text/javascript" src="url/file/js"> </script>`
- Nella sezione `<body>` creare un "contenitore" per il geobrowser:
  - `<div id="map3d" style="height: 400px; width: 600px;"></div>`



## API GOOGLE – GOOGLE.EARTH NAMESPACE REFERENCE

- Contiene funzioni globali che facilitano l'uso delle interfacce e delle funzionalità di queste API
  - `createInstance` (`object domNode`, `function initCallback`, `function failureCallback`, `[optional] object options`)
- Crea un'istanza del plugin nel nodo DOM passato come input (il contenitore che avete creato).
- Se la creazione ha successo viene eseguita la funzione `initCallback`.
- In caso contrario viene eseguita la funzione `failureCallback`.



## ESEMPIO CREATE INSTANCE



```
<script type="text/javascript">
  var ge;
  google.load("earth", "1");

  function init() {
    google.earth.createInstance('map3d', initCB, failureCB);
  }

  function initCB(instance) {
    ge = instance;
    ge.getWindow().setVisibility(true);
  }

  function failureCB(errorCode) {
  }

</script>
```



## API GOOGLE – GOOGLE.EARTH NAMESPACE REFERENCE

- `addEventListener` (`object targetObject`, `string eventID`, `function listenerCallback`,`[optional] bool useCapture`)
- Attiva un listener per l'evento *eventID* ( 'click', 'mouseover',etc) sull'oggetto *targetObject* che invoca la funzione *listenerCallback*.
- `removeEventListener` (`object targetObject`, `string eventID`, `function listenerCallback`,`[optional] bool useCapture`).
- Rimuove l'evento *eventID* dall'oggetto *targetObject*.



## ADDEVENTLISTNER EXAMPLE

```
//feat: some TargetObject
google.earth.addListener(feat, 'mouseover', showBalloon);
function showBalloon(event)
{
    var balloon = ge.createFeatureBalloon("");
    balloon.setMaxWidth(300);
    balloon.setFeature(event.getTarget());
    ge.setBalloon(balloon);
}
```



## API GOOGLE – GOOGLE.EARTH NAMESPACE REFERENCE

- `fetchKml` (`GePlugin pluginInstance`, `string url`, `function completionCallback`):
- Effettua il parsing del file kml indicato nell'url e restituisce l'oggetto KML.
- Può essere utilizzata anche quando GE non è stato ancora caricato.
- Per visualizzare il kml sul globo bisogna effettuare l'attach dell'oggetto restituito tramite il metodo `appendChild`



## FETCHKML EXAMPLE

....

```
google.earth.fetchKml(ge, kmlUrl,  
    finishFetchKml);
```

```
function finishFetchKml(kmlObject) {  
    // check if the KML was fetched properly  
    if (kmlObject) {  
        // add the fetched KML to Earth  
  
        ge.getFeatures().appendChild(kmlObject);  
    } else {  
        // setTimeout to prevent deadlock in  
        some browsers  
        setTimeout(function() {  
            alert('Bad or null KML.');        }, 0);  
    }  
}
```





## GEFUTURECONTAINER

- Contiene tutti gli elementi visualizzati sul geobrowser, permette di navigare tra gli elementi e di modificarli.
- Si invoca tramite il metodo del plugin: `getFeatures()`.
- Metodi:
  - `appendChild(KmlObject object)`
  - `getChildNodes()`
  - `getFirstChild()`
  - `getLastChild()`
  - `hasChildNodes()`
  - `insertBefore(KmlObject newChild, KmlObject refChild)`
  - `removeChild(KmlObject oldChild)`
  - `replaceChild(KmlObject newChild, KmlObject oldChild)`



## GEPLUGIN

- è la classe principale di queste API, è l'oggetto ritornato quando si crea un istanza. Offre dei metodi factory per costruire gli altri oggetti del geobrowser (plugin, ballon ecc.). I metodi principali sono:
  - `parseKml` (`string kml`)
  - effettua il parsing di una stringa KML creando un oggetto KML (`KMLObject`) e ne ritorna il puntatore
  - Anche in questo caso bisogna effettuare l'attach dell'oggetto ritornato.



## GEPLUGIN

- `createPlacemark` (string ID)
- crea un placemark, cioè un oggetto che ha una forma geometrica o un'icona, sul geobrowser.
- `createPoint` (string ID)
- Crea un punto sul geobrowser. Specifica il punto geografico definito da longitudine, latitudine e opzionalmente altitudine.
- Entrambe le funzioni restituiscono un oggetto vuoto, al quale bisogna settare le proprietà specifiche affinché sia visualizzabile.



# KMLOBJECT

- Classe astratta alla base di tutti gli oggetti create con le API del plug-in.
- `getType()`: restituisce il tipo dell'oggetto.
- `getID()`: restituisce l'identificativo univoco.
- `compareTo (KmlObject compare)`: restituisce un valore booleano a seconda che l'oggetto compare sia uguale o meno a quello su cui si invoca il metodo
- `getParentNode()`: restituisce il genitore dell'oggetto.



## KML PLACEMARK

- È un'istanziatura di `KMLObject` e gestisce i placemark sul geobrowser.
- Offre metodi per la gestione della geometria del placemark e per la navigazione nell'albero formato da questi oggetti.
- `getGeometry()`: restituisce il tipo di forma geometrica del placemark (punto, linea, etc).
- `setGeometry (KMLGeometry geometry)`: permette di modificare la geometria del placemark.
- `getKML()`: restituisce il KML che rappresenta l'oggetto.
- `getName()`: restituisce il nome dell'oggetto



# PAGINA WEB CON GOOGLE EARTH PLUGIN

```
o <html>
  <head>
    <title>Sample</title>
    <script src="http://www.google.com/jsapi?key=ABCDEFGH"> </script>
    <script type="text/javascript">
      var ge;
      google.load("earth", "1");

      function init() {
        google.earth.createInstance('map3d', initCB, failureCB);
      }

      function initCB(instance) {
        ge = instance;
        ge.getWindow().setVisibility(true);
      }

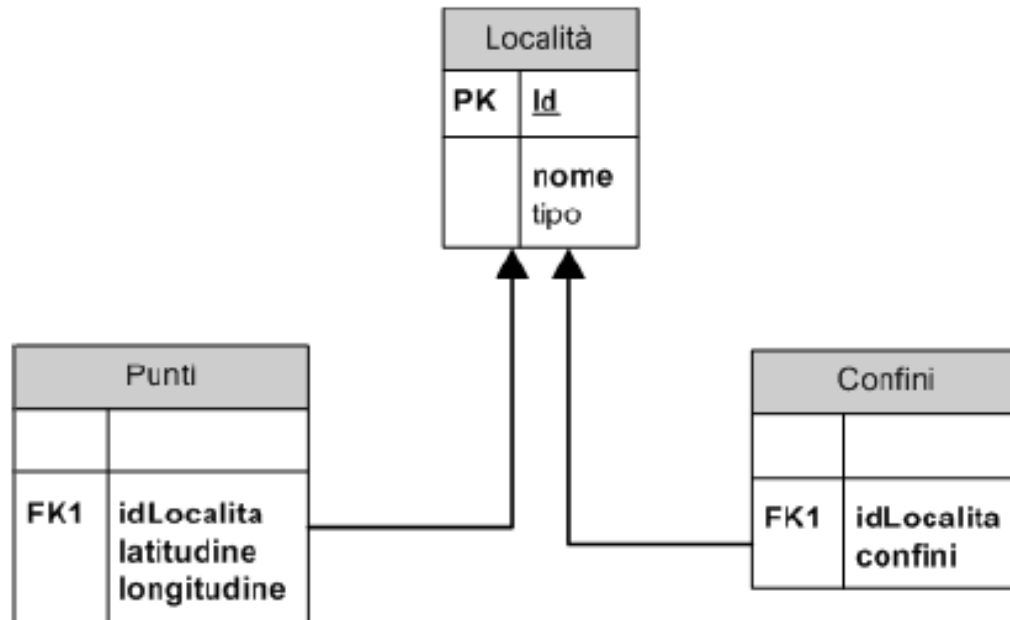
      function failureCB(errorCode) {
      }

      google.setOnLoadCallback(init);
    </script>

  </head>
  <body>
    <div id="map3d" style="height: 400px; width: 600px;"></div>
  </body>
</html>
```



# DB RELAZIONALE CON COORDINATE



## KML DI PUNTI

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
  <name>Potenza</name>
  <description>Potenza</description>
  <Point>
    <coordinates> 15.480893,40.381157 </coordinates>
  </Point>
</Placemark>
.....
</Document>
</kml>
```





## SUGGERIMENTI

- Ricordate che le icone sui placemark e le label possono essere visualizzate solo sulle geometry di tipo Point
- Per Visualizzare il percorso di un oggetto potete utilizzare una linea
- Se l'oggetto torna al museo di base, dovrete utilizzare un poligono.
- Se l'oggetto viene prestato più volte da un museo, avrete bisogno di più poligoni (multigeometry) o di un insieme di linee.
- Se due musei risiedono nella stessa città e vanno visualizzati contemporaneamente, per evitare sovrapposizioni aumentate o diminuite di qualche centesimo la latitudine e/o la longitudine.

