

# Primary Data Handling in the Geoweb: Alternate Structures of Data

GEOG 384



What is a Geographic Feature and  
How do we Structure it?

# How it started

Attributes of GW As Concentrations										
Shape	Staid	State	Fips	Latitude	Longitude	Use	Depth	Sample_da	Sample_tim	Conc_ppb
Point	333733087290302	AL	1057	33.626	-87.484		30	19820713	1300	0
Point	333737087310101	AL	1057	33.627	-87.517			19800528	830	1
Point	334131087344201	AL	1057	33.692	-87.578			19790927	1100	2
Point	334151087362001	AL	1057	33.698	-87.606			19790927	915	2
Point	334217087323602	AL	1057	33.705	-87.543			19820716	1345	1
Point	334220087324501	AL	1057	33.706	-87.546			19800515	845	1
Point	334250087323601	AL	1057	33.714	-87.543			19800515	1200	1
Point	334250087323701	AL	1057	33.714	-87.544		125	19820716	1215	1
Point	334250087323702	AL	1057	33.714	-87.544		101	19820716	1245	1
Point	334315087320001	AL	1057	33.721	-87.533			19791002	1200	3
Point	334330087332401	AL	1057	33.725	-87.557		20	19820716	900	0
Point	334330087332402	AL	1057	33.725	-87.557		90	19820716	915	2
Point	334440087325901	AL	1057	33.744	-87.550		44	19820715	1245	0
Point	334440087325902	AL	1057	33.744	-87.550		95	19820715	1300	1
Point	335110087342601	AL	1057	33.853	-87.574		124	19800513	1300	1
Point	335112087341201	AL	1057	33.853	-87.570			19800513	1230	1
Point	335141087343401	AL	1057	33.861	-87.576			19800603	1500	2
Point	310132085024001	AL	1069	31.026	-85.044	U	118	19950913	1250	0
Point	310549085264501	AL	1069	31.097	-85.446	H	144	19950913	1805	0
Point	311455085131701	AL	1069	31.249	-85.221	S	100	19950911	1435	0

# HTML

```
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

# XML

```
<student>
  <name>Renee Sieber</name>
  <address>
    <street>Milton</street>
    <city>Montreal</city>
  </address>
</student>
```

# JSON

```
{ "firstName": "Leon",
  "lastName": "Kennedy",
  "gender": "nb",
  "age": 28,
  "address": {
    "streetAddress": "123 Elm",
    "city": "Racoon City",
    "state": "IA",
    "postalCode": "39421"
  },
  "phoneNumbers": [
    { "type": "home", "number": "7383627627" }
  ]
}
```

**How it's going**

# Some notes about ML tags

They are keywords surrounded  
by angle brackets like

```
<placemark>
```

They tend to have content  
between tags

They tend to be paired (from  
KML)

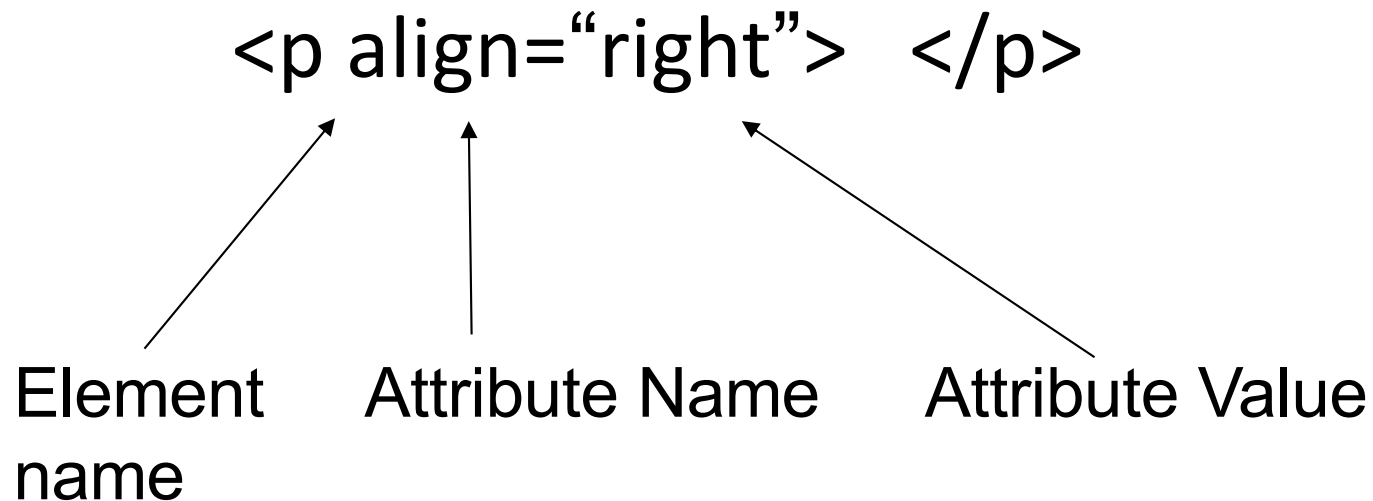
```
<placemark>                </placemark>
```

They're often nested

```
<placemark>  
    <Point>  
    </Point>  
</placemark>
```

The whole thing is called an  
element

# They may have attributes



`<hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>`

Sometimes  
the content is  
even less  
structured

```
<h1> Parc Lafontaine </h1>
```

```
<p>
```

```
The skating rink at the  
<i>parc</i> includes great  
skating, rental skates, and  
an outdoor café but  
<em>not</em> large size  
skates.
```

```
</p>
```

At least it  
looks less  
structured

```
<h1> Parc Lafontaine </h1>
```

```
<p>
```

The skating rink at the

```
<i>parc</i>
```

includes great  
skating, rental skates, and  
an outdoor café but

```
<em>not</em>
```

large size skates.

```
</p>
```



# HTML

```
<html>
  <head>
    <title>Hello
World!</title>
  </head>
  <body>
    <h1>My First
Heading</h1>
    <p>My first
paragraph.</p>
  </body>
</html>
```

# GeoJSON

GeoJSON is a well-specified standard, while a generic JSON file need not follow any standard for data organization.

# Components

Feature types

Feature order: array of  
coordinates in order

[longitude, latitude,  
elevation]

Properties

You can define!

It's render-specific!

# GeoJSON: It's Syntax, not Rocket Science

```
{  
  "type": "Feature",  
  "geometry": {  
    "type":  
      "GeometryCollection",  
    "geometries": [{  
      "type": "Point",  
      "coordinates": [0, 0]  
    }, {  
      "type": "LineString",  
      "coordinates": [[0, 0], [1, 0]]  
    }  
  ],  
  "properties": {  
    "name": "null island"  
  }  
}
```

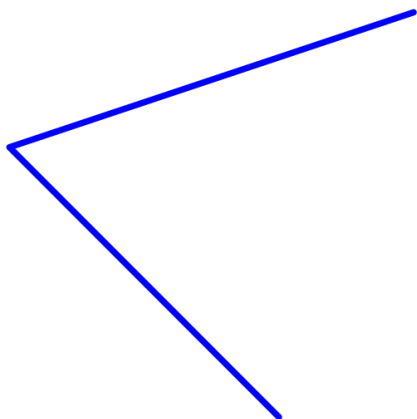
# Polygons

- { "type": "LineString",  
"coordinates": [[0, 0], [10, 10]] }
- { "type": "Polygon",  
"coordinates": [[ [0, 0], [10, 10], [10, 0], [0, 0] ] ] }
- Innies and outies!
- Storing styles and attributes

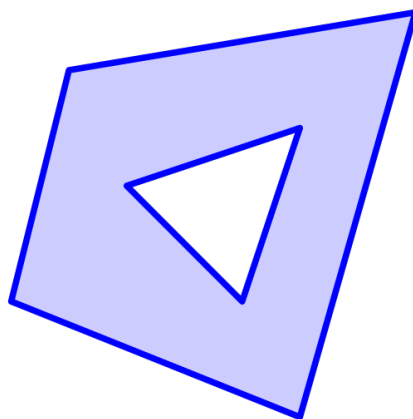
**Point**



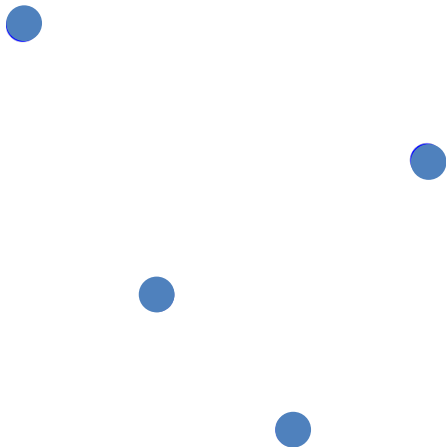
**LineString**



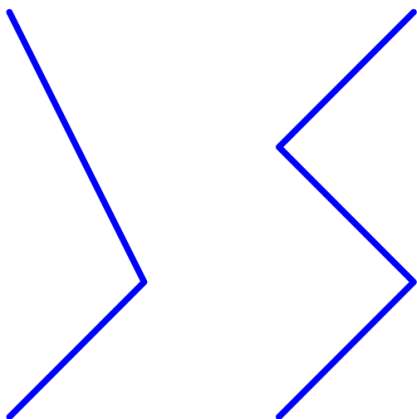
**Polygon**



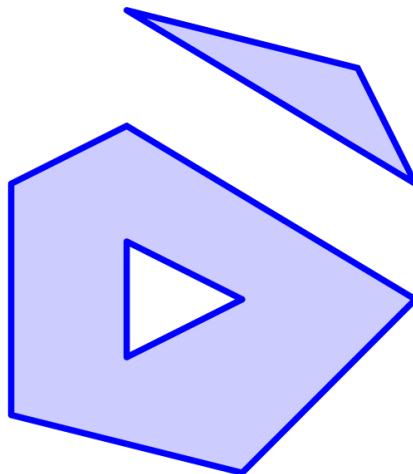
**MultiPoint**



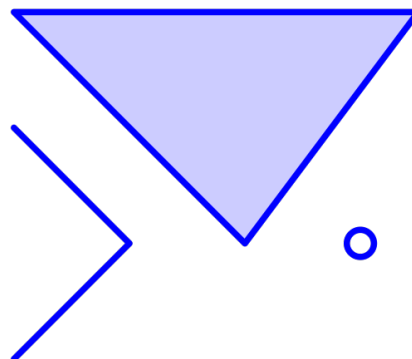
**MultiLineString**



**MultiPolygon**



**GeometryCollection**



# Properties

```
{ "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [0, 0]  
  }, "properties": {  
    "name": "null island"  
  }  
}
```



## Polygons

Even more complicated when polygons are jurisdictional