

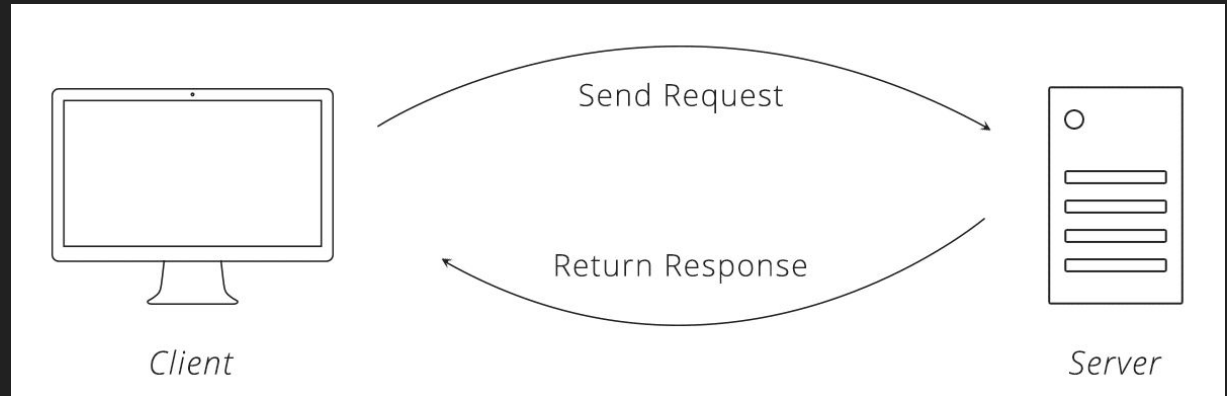
# APIs

# Useful APIs

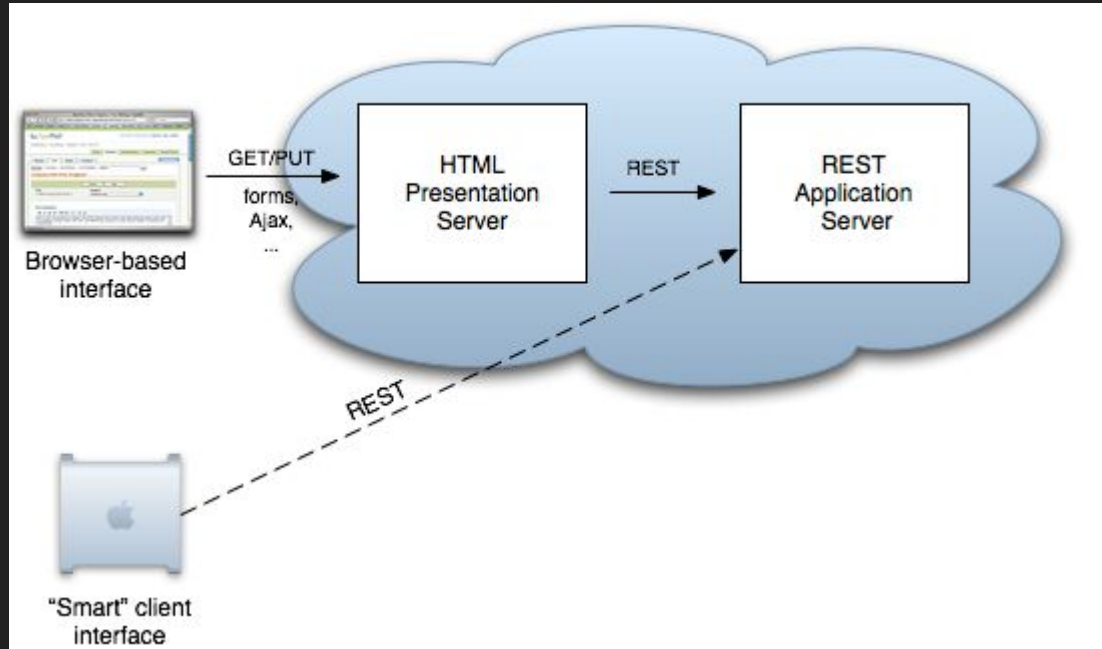
- Google APIs (for everything).
- Google Analytics.
- Twitter API.
- Facebook API.
- eBay API.

# Introduction to HTTP

- Server
- Request
- Response



# REST - Representational State Transfer



# API - Application Program Interface

- How a program can interact with an application.
- Specifications for authentication and more.
- Principles of RESTful APIs.



# Sending HTTP Request

```
var xhr = new XMLHttpRequest();  
xhr.open("GET", "https://www.codecademy.com/", false);  
xhr.send();  
  
console.log(xhr.status);  
console.log(xhr.statusText);
```

# Sending an HTTP Request

```
var theUrl = "https://www.codecademy.com";  
var xhr = new XMLHttpRequest();  
xhr.onreadystatechange = function() {  
    console.log(xhr.status);  
}  
xhr.open("GET", theUrl, true);  
xhr.send();
```

# Sending an HTTP Request

```
function httpGetAsync(theUrl)
{
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        console.log(xhr.status);
    }
    xhr.open("GET", theUrl, true);
    xhr.send(null);
}

httpGetAsync('https://www.codecademy.com');
```



# Sending an HTTP Request

```
function httpGetAsync(theUrl, callback)
{
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        callback(xhr.status);
    }
    xhr.open("GET", theUrl, true);
    xhr.send(null);
}
```

```
httpGetAsync('https://www.codecademy.com/doesntexist', function(status){
    console.log(status);
});
```

# HTTP Methods

- GET - Retrieves information from server
- POST - Sends information to specified source
- PUT - Updates existing information to specified source
- DELETE

# HTTP Request

- Request: what kind of request it is
- Header: additional information like which client makes the request.

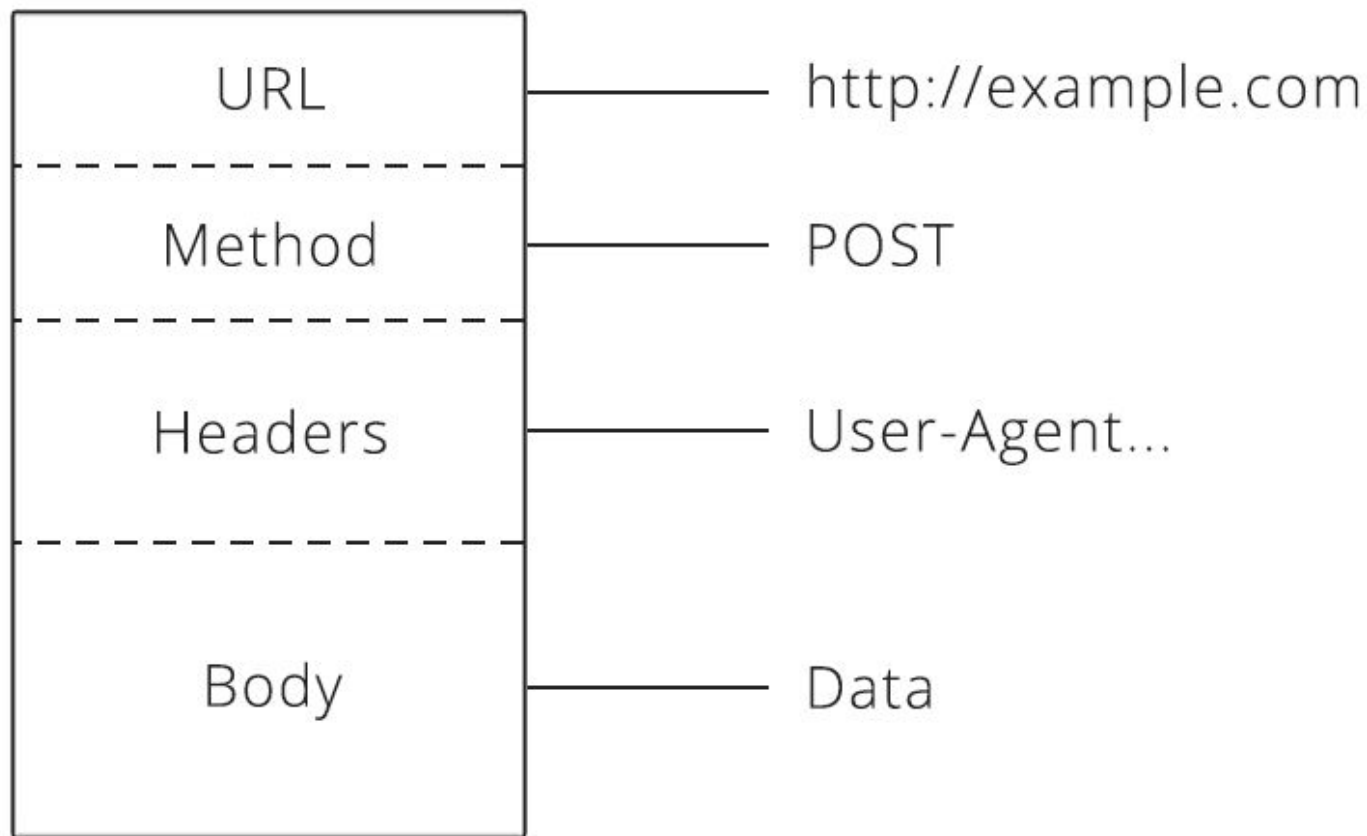
- Body

```
POST /google/blog HTTP/1.1
```

```
Host: www.google.com
```

```
Content-Type: text/html; charset=UTF-8
```

```
Name=Omar&Age=19
```



*Request*

# API Authentication

- API keys are normally needed

<https://developers.google.com/maps/documentation/javascript/get-api-key>



# HTTP Response

1xx

2xx

3xx

4xx

5xx

# 404 web error

## **Not Found**

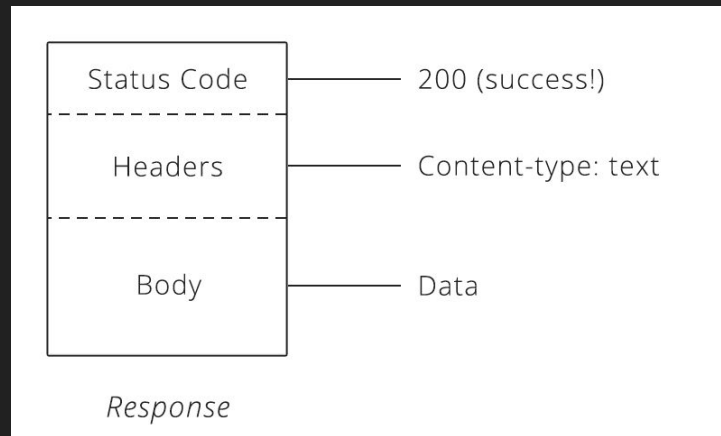
The requested URL / was not found on this server.

---

*Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4 with Suhosin-Patch Server*

# HTTP Response

- Response: HTTP Status code
- Header: Further info



- Body

**HTTP/1.1 200 OK**

**Content-Type: text/xml; charset=UTF-8**

**<?xml version="1.0" encoding="utf-8"?>**

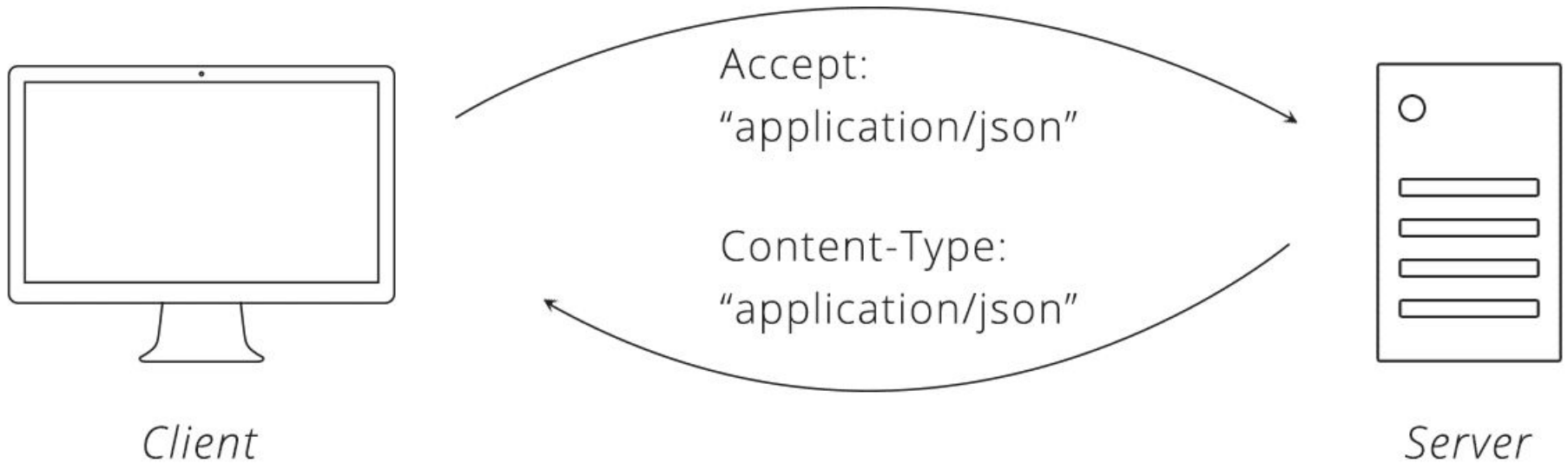
**<string xmlns="https://www.codecademy.com/">Accepted</string>**



# XML

```
<user>  
  <name>Osansevier</name>  
  <age>19</age>  
  <carreer>Developer</carreer>  
</user>
```

# Content-type



# Parsing XML

```
xhr.setRequestHeader('Content-Type', 'text/xml');  
xhr.send();  
  
xmlDocument = xhr.responseXML;  
console.log(xmlDocument.childNodes[0].textContent);
```

# JSON

```
{ "courses" : [  
  {  
    "name" : "Web Development",  
    "hours" : 300  
  },  
  {  
    "name" : "Cooking",  
    "hours" : 3023  
  }  
]  
}
```

# String to JSON

```
var demo = '{"pets": { "name": "Jeffrey", "species": "Giraffe"}}';  
  
var json = JSON.parse(demo);  
console.log(json);
```

# XML vs JSON

- Read documentation
- Each one has its own advantages

# OAuth 2

- User
- Client
- Server

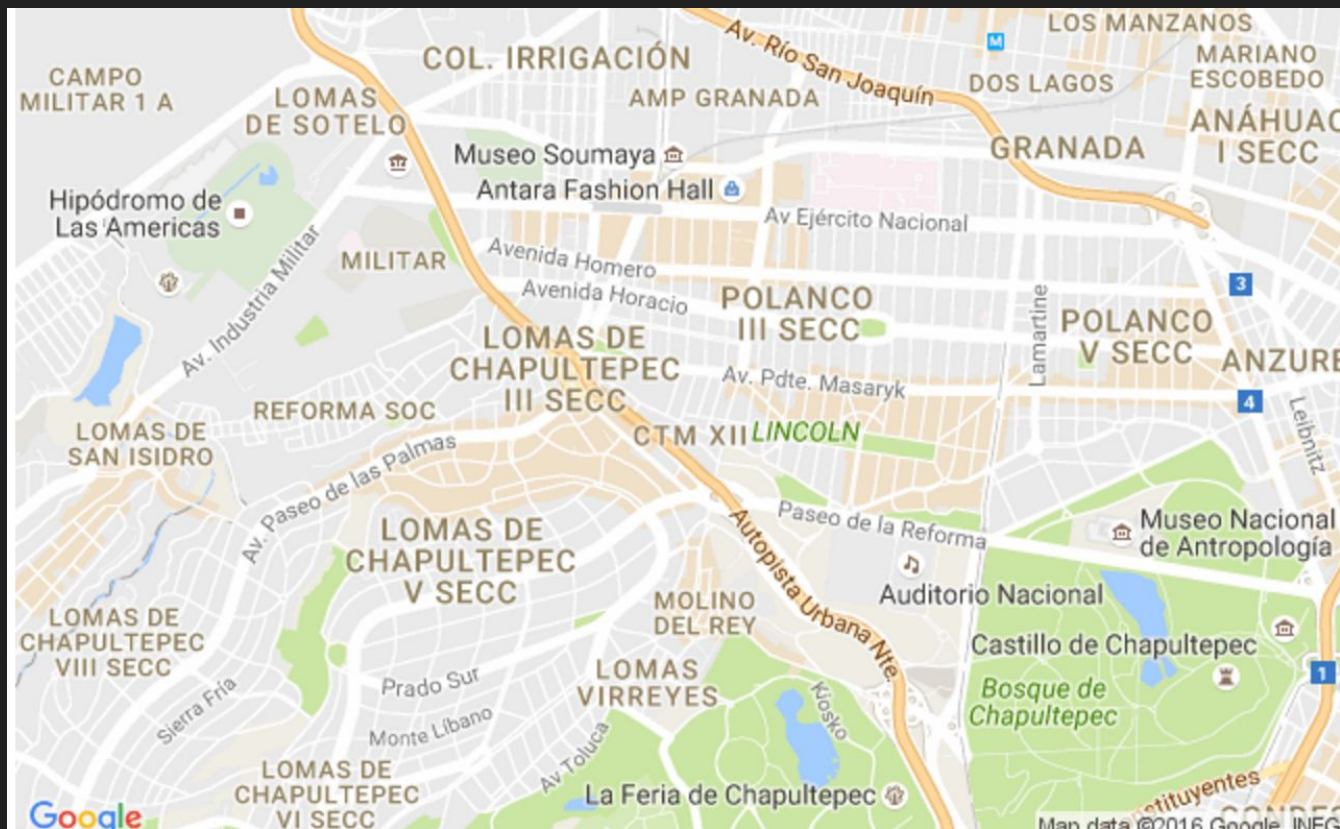
# Geolocation

```
navigator.geolocation.getCurrentPosition(showPosition);
```

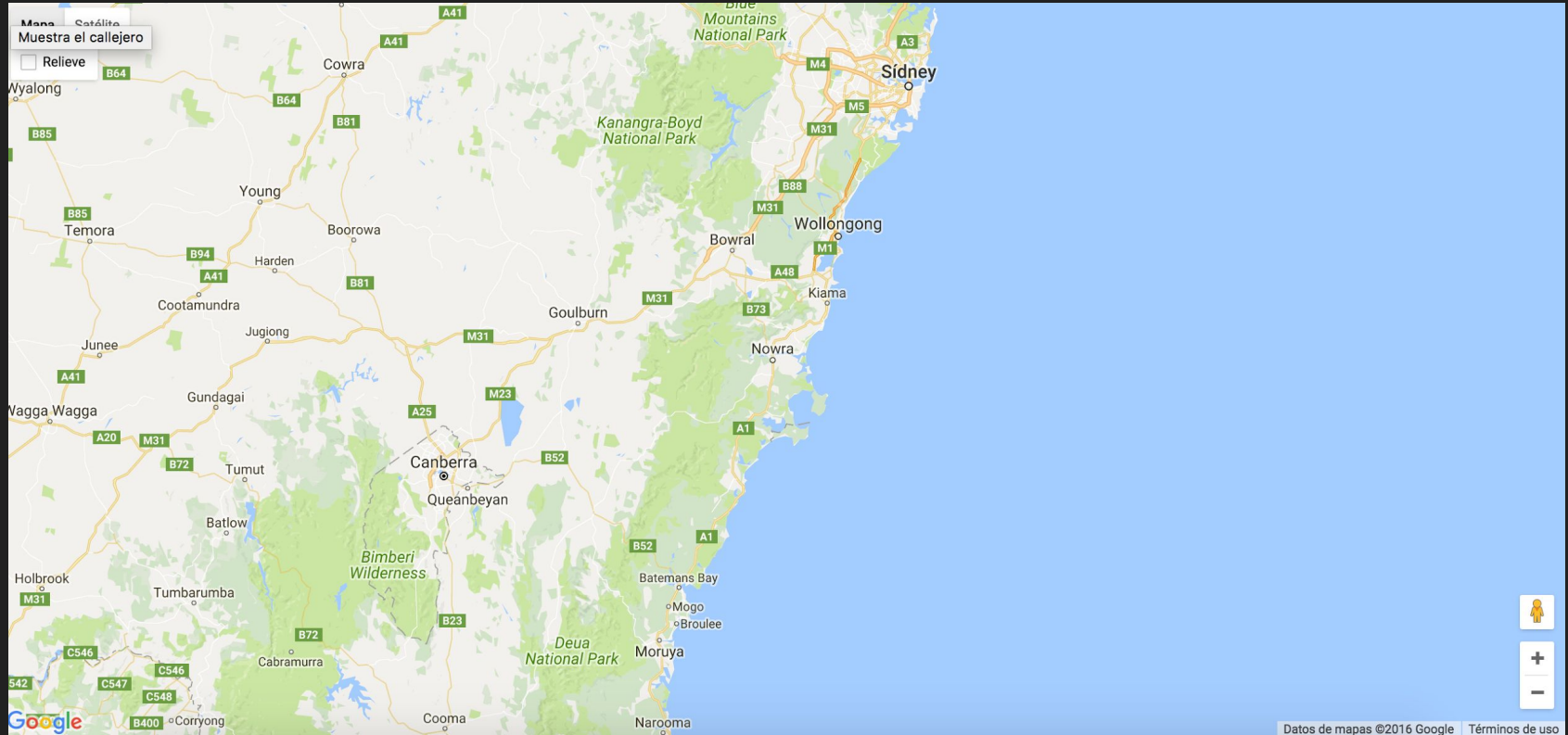
```
function showPosition(position) {  
    console.log(position.coords.latitude);  
    console.log(position.coords.longitude);  
}
```



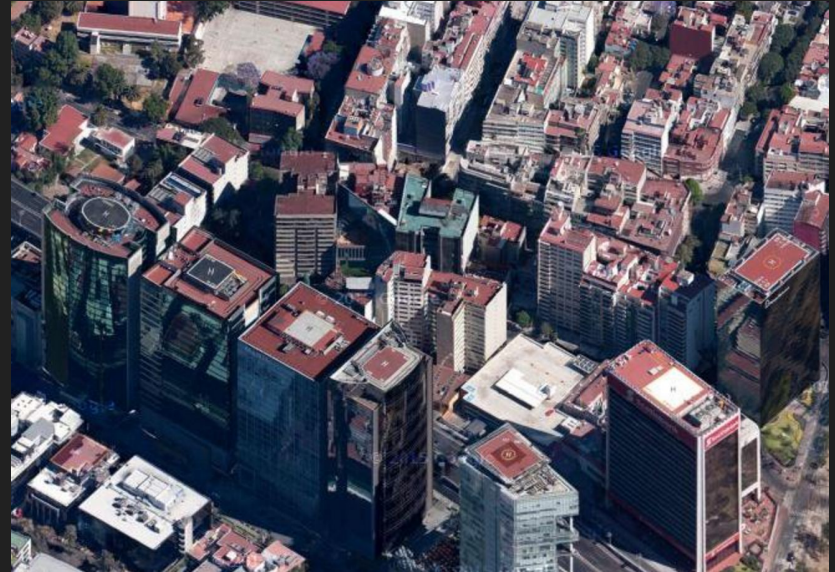
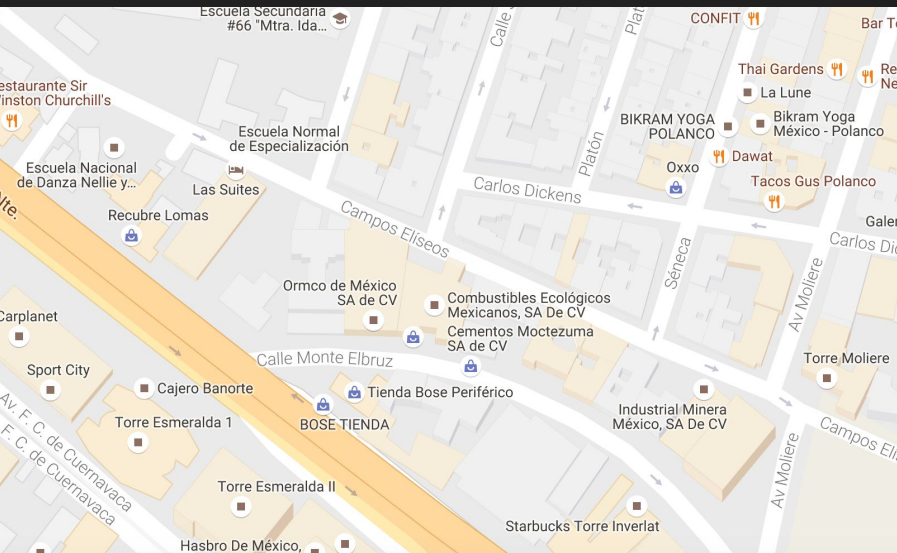
# Static Maps



# Maps (hardcoded locations)



# Finally! Maps



# Reverse geocoding

```
function getReverseGeocodingData(lat, lng) {  
    var latlng = new google.maps.LatLng(lat, lng);  
    var geocoder = new google.maps.Geocoder();  
    geocoder.geocode({ 'latLng': latlng }, function (results, status) {  
        console.log(results);  
        var address = (results[0].formatted_address);  
    });  
}
```



# Weather API

