API Google MAPS

Google API

Google API es un conjunto de APIs desarrolladas por Google, que permiten la comunicación con los servicios de Google y su integración con otros servicios. Ejemplos de estos incluyen Search, Gmail, Translate o Google Maps.

¿Cómo funciona Google Maps?

Es sólo HTML, CSS y JavaScript trabajando conjuntamente. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un en la página HTML.

El API consiste de archivos JavaScript que contienen las clases, métodos y propiedades que se usan para el comportamiento de los mapas.

¿Cómo se usa?

Las coordenadas están expresadas usando números decimales separados por coma. La latitud siempre precede la longitud. La latitud es positiva si es N y negativa si es S. La longitud es positiva si es E y negativa si es O.

En los mapas físicos, las coordenadas están expresadas en grados, la posición de Benidorm es:

38° 32' 03. N, 0° 07' 53. O

La forma de convertir estos datos a decimales es:

(38° 32' 03. N) = (38 + (32 / 60) + (03 / 3600)) = 38,534167

(0° 07' 53. O) = -(0 + (07 / 60) + (53 / 3600)) = -0,131389

La longitud se multiplica por negativo, porque está a la izquierda (oeste) del punto 0,0.

Google estableció en varios métodos que la mayor cantidad a trabajar es 6 decimales. Es decir, cuando vayamos a establecer los decimales se puede hacer así:

- 5 a 6 decimales: es el máximo que debemos usar para ser específicos.
- 4 decimales: para algún detalle en el mapa.
- 3 decimales: es bueno para centrar ciudades.
- 2 decimales: es apropiado para centrar países o estados, tal vez 3 para países pequeños.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="googleMaps.css">
    <title>Prova Google Maps API</title>
    <script type="text/javascript" src="http://maps.google.com/maps/api/js?
    sensor=false&amp;language=es"></script>
    <script type="text/javascript" src="./js/map.js"></script>
</head>
<body>
    <div id="map"></div>
</body>
</html>
```

```
*{
    margin: 0;
    padding: 0;
}
html, body, #map{
    width: 100%;
    height: 100%;
}


function mapaGoogle(){
    var options = {
        zoom: 8,
        center: new google.maps.LatLng(38.53, -0.13) ,
        mapTypeId: google.maps.MapTypeId.SATELLITE
    };
    var map =
    new google.maps.Map(document.getElementById('map'), options);
};
window.addEventListener('load',mapaGoogle,false)
```

En el siguiente código HTML, que carga los scripts de javascript:

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false&;amp;language=es">
</ script>
```

La dirección apunta al API, pero también se requiere pasar una cadena de consulta con la clave ***sensor***. Esto le indica a Google si el dispositivo que usa el mapa, tiene un dispositivo que determina la geolocalización, como por ejemplo un GPS. Es obligatorio mencionarlo indicando como valor si es *false* o *true*. Falso para los que no lo usan y cierto para los que sí lo usan. Esto lo usa Google para proveer estadísticas a sus proveedores. No tiene que ver con habilitarlo para la geolocalización.

También se añadió el lenguaje, aunque el API trata de determinar cuál es el lenguaje a mostrar, pero se puede especificar usando la clave ***language***.

Es importante que el siguiente código…

```
<script type="text/javascript" src="map.js"></script>
```

…se encuentre debajo del elemento **<script>** que incluye el API, para que así se puedan cargar todas las clases, métodos y propiedades a usar. Las clases, métodos y propiedades comienzan con google.maps. A esto se le conoce como namespace.

MapOptions

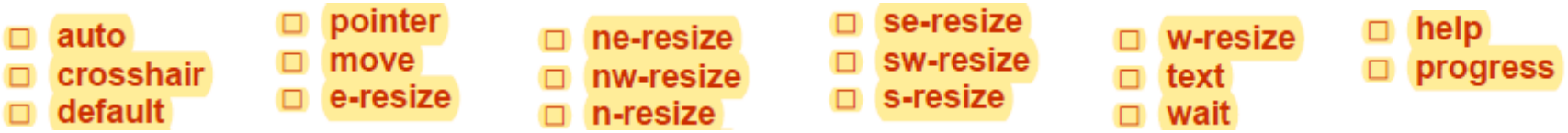
MapOptions contiene la información de cómo queremos ver el mapa y cómo queremos que se comporte. Hay tres propiedades que son **obligatorias**:

- **zoom:** Define el "zoom" inicial. Debe ser un número entre el 1 y el 23. El 1 es el mapa totalmente alejado y 23 es lo más cercano posible.
- **center:** Define el centro del mapa con las coordenadas. Las coordenadas deben indicarse usando el método ***google.maps.LatLng(latitud, longitud).***
- **mapTypeId:** Define qué tipo de mapa se desea mostrar al inicio.

Pero hay muchas otras propiedades opcionales para **MapOptions**:

- **zoomControl:** muestra botones "+" i "-" para cambiar el nivel de zoom del mapa. Aparece por defecto en la esquina inferior derecha del mapa.
  - Valores: true|false, por defecto false.
- **rotateControl:** proporciona una combinación de inclinación y opciones de rotación de mapas que contienen imágenes oblicuas. Solo estará disponible si en la localización existen imágenes en 45º. Este control aparece de forma predeterminada en la parte inferior derecha del mapa.
  - Valores: true|false, por defecto true.

- **keyboardShortcuts**: Habilita o inhabilita el uso del teclado. Las teclas a usar son las flechas para mover el mapa y +/- para el “zoom”.
  - Valores: true|false, por defecto true.
- **disableDoubleClickZoom**: Habilita o inhabilita el doble click del ratón para hacer “zoom”.
  - Valores: true|false por defecto false.
- **draggable**: Habilita o inhabilita el poder arrastrar el mapa.
  - Valores: true|false, por defecto true.
- **scrollwheel**: Habilita o inhabilita el poder hacer “zoom” con la rueda del ratón.
  - Valores: true|false, por defecto true.
- **draggableCursor**: Indica el tipo de cursor a mostrar cuando el ratón está encima del mapa. El valor es de tipo cadena. Puede ser de los de una computadora por defecto y la mayoría de los que están mencionados en el siguiente listado. También puede ser uno personalizado y la ruta puede ser local (en el servidor de la aplicación) o una dirección web externa.



- **draggingCursor**: Indica el tipo de cursor a mostrar cuando el ratón está presionado en el mapa. El valor es del tipo cadena y pueden ser los mismos tipos que en el atributo anterior.

- **backgroundColor**: Esta propiedad afecta el color del fondo del contenedor. Típicamente se ve cuando se arrastra el mapa o cuando carga al inicio. Puedes usar un valor hexadecimal o la forma estándar (red, yellow, green, blue, etc). Por defecto el color es con el valor hexádecimal #E5E3DF.
- **noClear**: Habilita o inhabilita que se sobreescriba lo que haya en el contenedor. Normalmente la forma como se trabaja para colocar contenido encima del contenedor es usando un elemento fuera del contenedor que muestra el mapa y con CSS se coloca en el lugar deseado en el mapa.
  - Valores: true|false, por defecto false.
- **disableDefaultUI**: Habilita o inhabilita mostrar el UI que viene predefinido.
  - Valores: true|false, por defecto false.
- **mapTypeControl**: Habilita o inhabilita el control de tipo de mapa.
  - Valores: true|false, por defecto true.
- **mapTypeControlOptions**: Opciones de visualización iniciales de control de tipo de mapa.
  - Valores de tipo: (ROADMAP | SATELLITE | HYBRID | TERRAIN)
- **navigationControl**: Habilita o inhabilita el control de navegación.
  - Valores: true|false, por defecto true.

- **navigationControlOptions**: Opciones de visualización iniciales del control de navegación.
- **scaleControl**: Habilita o inhabilita el control de escala.
  - Valores: true|false, por defecto true.
- **scaleControlOptions**: Son las opciones de visualización iniciales del control de escala.
- **streetViewControl**: Habilita o inhabilita el hombrecito de “Street View”. Está disponible en ciertas áreas.
  - Valores: true|false, por defecto false.

```
var options = {
    zoom: 8,
    center: new google.maps.LatLng(38.53, -0.13),
    mapTypeId: google.maps.MapTypeId.SATELLITE,
    backgroundColor: '#ffffff',
    noClear: true,
    disableDefaultUI: true,
    keyboardShortcuts: false,
    disableDoubleClickZoom: true,
    draggable: false,
    scrollwheel: false,
    draggableCursor: 'move',
```

```
        draggingCursor: 'move',
        mapTypeControl: true,
        mapTypeControlOptions: {
            style: google.maps.MapTypeControlStyle.HORIZONTAL_MENU,
            position: google.maps.ControlPosition.TOP_LEFT,
            mapTypeIds: [ google.maps.MapTypeId.SATELLITE ]
        },
        navigationControl: true,
        streetViewControl: true,
        navigationControlOptions: {
            position: google.maps.ControlPosition.TOP_RIGHT,
            style: google.maps.NavigationControlStyle.ANDROID
        },
        scaleControl: true, scaleControlOptions: {
            position: google.maps.ControlPosition.TOP_LEFT,
            style: google.maps.ScaleControlStyle.DEFAULT
        }
};
```

Modificar los valores ya asignados con setOptions

Hasta ahora, hemos iniciado los valores directamente en el **MapOptions** para iniciar el mapa. Sin embargo, después de que se hayan cargado, podemos modificarlos con el método **setOptions**, que es de gran utilidad para interactuar con el usuario.

Existen tres que no son modificables: ***noClear***, ***backgroundColor*** y ***disableDefaultUI***

```
map.setOptions({
    zoom: 10,
    center: new google.maps.LatLng(38.53, -0.13),
    mapTypeId: google.maps.MapTypeId.TERRAIN,
    keyboardShortcuts: true,
    disableDoubleClickZoom: false,
    draggable: true,
    scrollwheel: true,
    draggableCursor: 'hand',
    draggingCursor: 'hand',
    mapTypeControlOptions: {
        style: google.maps.MapTypeControlStyle.DROPDOWN_MENU,
        position: google.maps.ControlPosition.TOP_RIGHT,
```

```
            mapTypeIds: [ google.maps.MapTypeId.ROADMAP,
            google.maps.MapTypeId.SATELLITE ]
        },
        navigationControlOptions: {
            position: google.maps.ControlPosition.TOP_LEFT,
            style: google.maps.NavigationControlStyle.ZOOM_PAN
        },
        scaleControlOptions: {
            position: google.maps.ControlPosition.BOTTOM_LEFT,
            style: google.maps.ScaleControlStyle.DEFAULT
        }
});
```

Getters y Setters

Existen métodos para modificar y/o obtener los resultados de las **propiedades obligatorias**.

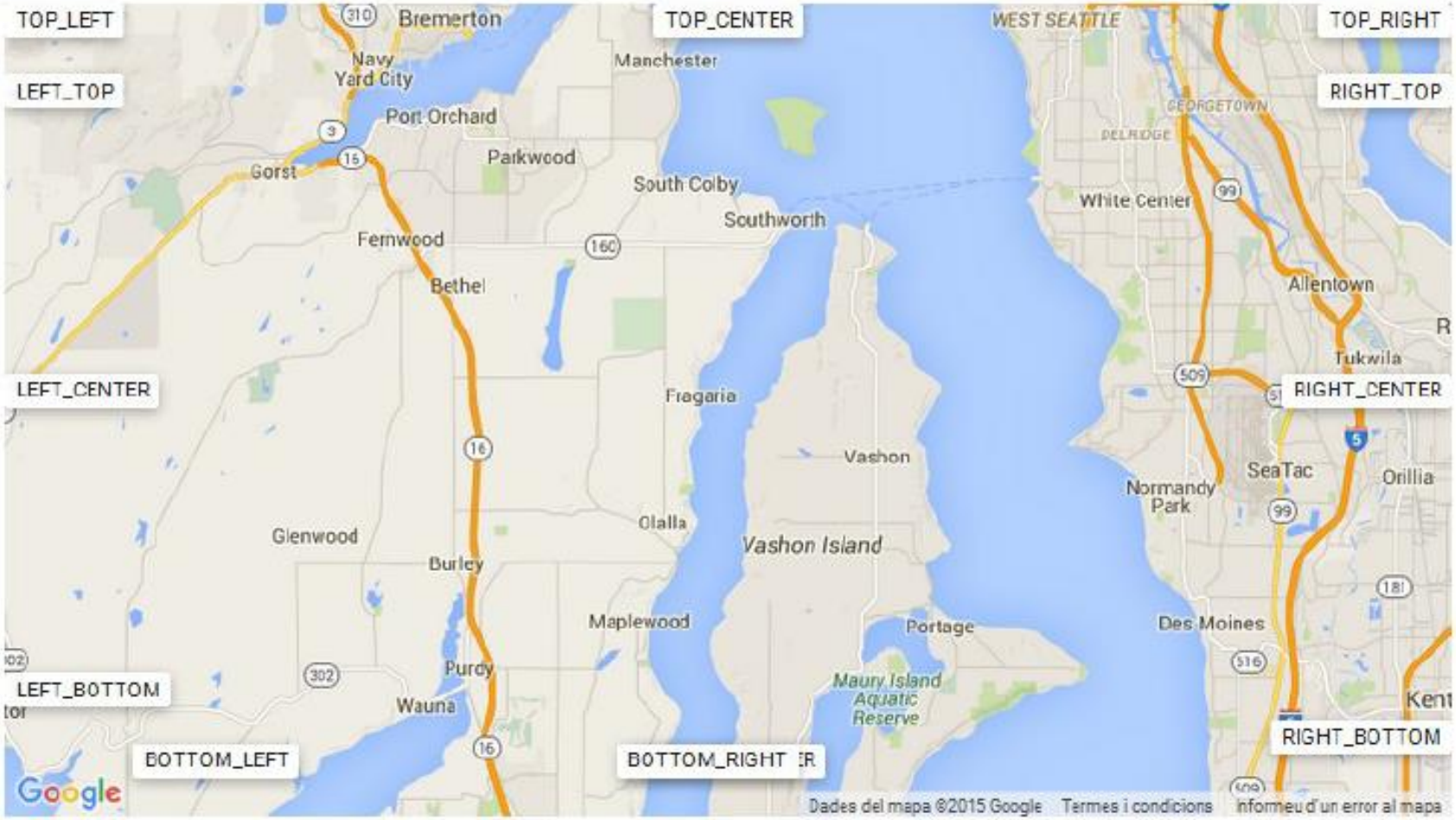
- getZoom() y setZoom(*1-23*)
- getCenter() y setCenter(*google.maps.LatLng(latitud, longitud)*)
- getMapTypeId() y setMaptTypeId(*google.maps.MapTypeId.**)

Por Ejemplo:

```
map.setZoom(9);
var zoomLevel = map.getZoom();
map.setCenter(new google.maps.LatLng(18.17, -66.3));
var centerOfMap = map.getCenter();
map.setMapTypeId(google.maps.MapTypeId.ROADMAP);
var mapTypeIdOfMap = map.getMapTypeId();
```

Posicionando los elementos

En las opciones de los controles podemos ubicarlos en las siguientes posiciones



- **TOP_CENTER** debe ser colocado a lo largo del centro de la parte superior del mapa.
- **TOP_LEFT** indica que el control debe ser colocado en la parte superior izquierda del mapa.
- **TOP_RIGHT** indica que el control debe ser colocado en la parte superior derecha del mapa.
- **LEFT_TOP** indica que el control debe ser colocado en la parte superior izquierda del mapa, pero debajo de los elementos TOP_LEFT.
- **RIGHT_TOP** indica que el control debe ser colocado en la parte superior derecha del mapa, pero por debajo de cualquier elemento TOP_RIGHT.
- **LEFT_CENTER** el control debe ser colocado en el centro del lado izquierdo del mapa.
- **RIGHT_CENTER** el control debe ser colocado centrado en la parte derecha del mapa.
- **LEFT_BOTTOM** indica que el control debe ser colocado en la parte inferior izquierda del mapa, pero por encima de cualquier elemento BOTTOM_LEFT.
- **BOTTOM_LEFT** indica que el control debe ser colocado en la parte inferior izquierda del mapa, hacia la parte inferior central.
- **RIGHT_BOTTOM** indica que el control debe ser colocado en la parte inferior derecha del mapa, pero por encima de cualquier elemento BOTTOM_RIGHT.
- **BOTTOM_RIGHT** indica que el control debe ser colocado en la parte inferior derecha del mapa, hacia la parte inferior central.
- **BOTTOM_CENTER** el control debe ser colocado en el centro de la parte inferior del mapa.

Añadir un marcador

Un marcador identifica una ubicación en un mapa. Por defecto, un marcador utiliza una imagen estándar. Los marcadores pueden mostrar imágenes personalizadas, conocidos generalmente como "iconos". Los marcadores y los iconos son objetos de tipo **Marker**. Se puede configurar un icono personalizado dentro del constructor, o llamando **setIcon()** en el marcador.

El constructor **google.maps.Marker** toma como parámetro un solo objeto de opciones que especifica las propiedades iniciales del marcador.

Los siguientes campos son comúnmente establecidos en la construcción de un marcador:

- **position** (obligatorio) especifica un LatLng que indica la localización inicial del marcador.
- **map** (opcional) especifica el mapa en el que colocar el marcador. Si no se especifica el mapa, se crea el marcador pero no aparece en el mapa. Se puede añadir el marcador más tarde llamando al método **marker.setMap(map)**

```
var marker = new google.maps.Marker({
    position: myLatLng,
    map: map,
    title: 'Benidorm'
});
```

Cambiar Icono del Marker

Simplemente se debe cargar una imagen (si es posible con un tamaño adecuado, 32x32px por ejemplo) en una variable en el código javascript y asignarlo al parametro *icon* del objeto *Marker*.

```
var imatge = './img/markerComputer.png';
var marker = new google.maps.Marker({
    position: {lat: 38.553437, lng: -0.120647},
    map: map,
    icon: imatge,
    draggable: true,
    animation: google.maps.Animation.DROP,
    title: 'IES Pere Maria Orts'
});
```

Si queremos insertar otro *marker*, debemos repetir el código con los nuevos parámetros:

```
var imatge2 = 'img/markerBall.png';
var marker = new google.maps.Marker({
    position: {lat: 38.549558, lng: -0.120647},
    map: map,
```

```
        icon:imatge2,
        draggable: true,
        animation: google.maps.Animation.DROP,
        title: 'Poliesportiu L'illa de Benidorm'
    });
```

También podemos definir características más concretas de la imagen que hará de icono:

- **url**: fuente del fichero de imagen.
- **size**: medida de la imagen original.
- **scaledSize**: medida a la que queremos que se escale una imagen (por ejemplo, si es demasiado grande para el mapa).
- **origin**: posición dentro de la imagen.
- **anchor**: ancla de la imagen (la posición que irá sobre el mapa).

```
var imagen = {
    url: './img/markerBlue.png',
    scaledSize: new google.maps.Size(20, 32),
    origin: new google.maps.Point(0, 0),
    anchor: new google.maps.Point(10, 32)
};
```

Ventana de información al marker

Con la función **infowindow()** se puede añadir una ventana de información cuando se hace clic sobre el *marker*. El contenido de la ventana, es mejor si se formatea con HTML.

```
var contenido = '<div íd="content" style="background-color:#DDD;">'+
    '<h1 >Ventana de información</h1>'+
    '<p style="text-
    align:center;padding:10px;margin:10px;">Información marker</p>'+
    '<p style="text-align:center;padding:10px;margin:10px;">+
    '<a href="http://www.link.com">'+
    '<img src="imagen.jpg" alt="imagen" width="200"/></a></p>'+
    '</div>'+
    '</div>';

var infowindow = new google.maps.InfoWindow({
    content: contenido
});

marker.addListener('click', function() {
    open(map, marker);
});
```


IES Pere Maria Orts i Bosch
Partida del Salt de l'Aigua
Carrer Barca del Bou, 9
Benidorm
Nivells: ESO / Batxillerat / FPB / Cicles Formatius
Lloc Web