

UNIVERSIDAD DE ROUEN

MASTER 1 GIL - 2010/2011

---

## Manual de utilización - HTML5Charts

---

*Equipo de proyecto*

Pierre COLLIGNON

Ludovic THUEUX

Maxence LUCE

Lyes KIMOUCHE

Radu POPICA

Abdourahmane DJIGO

*Professor*

Betsy DECAENS

# Índice general

<b>1. Una biblioteca de diagramas en HTML5</b>	<b>2</b>
1.1. Los diagramas . . . . .	2
<b>2. Utilización simple de la biblioteca</b>	<b>4</b>
2.1. Creación de origen de datos interna . . . . .	4
2.2. Creación de un diagrama . . . . .	4
2.3. Ejemplo . . . . .	4
<b>3. Utilización avanzada de la biblioteca</b>	<b>5</b>
3.1. Utilización de origen de datos externas . . . . .	5
3.2. Utilización de conectores . . . . .	5
3.3. Estilización . . . . .	5
3.3.1. Descripción del XML de estilización . . . . .	6
3.3.2. Utilización de fuentes internas y externas de estilización . . . . .	6

# Capítulo 1

## Una biblioteca de diagramas en HTML5

El producto *HTML5Charts* es una biblioteca JavaScript que permite el diseño de diagramas representativos de un conjunto de datos coherente. La automatización del tratamiento de los datos respecto de la colocación es completamente personalizable por medio de las opciones proporcionadas al usuario de la biblioteca, o por la re-implementación de los interfaces previsto a este respecto.

Este producto ha sido concebido para probar la viabilidad de un remplazo progresivo de Flash por HTML5.

### 1.1. Los diagramas

cuatro tipos de diagramas son disponibles :

- El histograma
- El histograma con efecto 3D
- El diagrama circular
- El diagrama en línea

Para cada uno de estos componentes dos representaciones son posibles, ya que los datos están modela por una tabla de doble entradas.

Podemos entonces diseñar en función de las filas o de las columnas.

Los diseños serán diferentes y la elección entre ambos se realiza en función de los datos que el usuario quiere resaltar.

Un ejemplo que podría ilustrar este principio sería el de una sociedad que quisiera realizar un análisis del número de compradores por cada producto que vende y por cada sucursal. Podría querer considerar el número de ventas de una sucursal para cada producto, poniendo en primer plano la comparación entre cada producto en el interior de una sucursal, o bien podría considerar el número de ventas de un producto por cada sucursal, poniendo entonces en primer plano la comparación entre cada sucursal para un producto dado.

Esta biblioteca será utilizada por 2 categorías de usuarios, la persona que quiera integrar un diagrama en su web y la persona que observa el diagrama en dicha web. Llamaremos respectivamente a uno y otro, el desarrollador y el internauta.

Para el desarrollador, como dijimos previamente, el objetivo es colocar un diagrama en su web. Para llevarlo a cabo dispondrá de nuestra biblioteca, que contiene los cuatro diagramas así como otras herramientas que explicaremos más adelante.

Esta persona transmitirá sus datos en forma de archivo XML, respetando los límites de un fichero XSD proporcionado, aplicable de manera independiente del diagrama deseado. Esta especificación de datos podrá ser llevada a cabo de dos maneras diferentes: escribir directamente en la página de HTML -llamada en interno-, o separarla de la página en un archivo cuyo camino estaría especificado -llamado en externo-. El estilo - el color de los diseños, la posición de la leyenda- podrá ser detallado de la misma manera que los datos pero independientemente.

Todas las combinaciones son posibles: los datos pueden ser dados en interno y el estilo en externo y a la inversa, los dos en interno o los dos en externo.

Para el internauta el diagrama ya está integrado en una web que visita. Él también juega un papel en estos diagramas (excepto en el diagrama en línea) puesto que puede interaccionar con ellos. Cuando pasa el cursor sobre uno de los datos (una barra en el histograma o sobre un sector del diagrama circular) este será resaltado por un cambio en el color. Esto permitirá poner de relieve un dato concreto. Además la información será completada por un recuadro con el nombre del dato así como su valor exacto. Este sistema hace aparecer un rectángulo que incluye el texto y que sigue al cursor mientras se encuentra dentro del dato.

## Capítulo 2

# Utilización simple de la biblioteca

La biblioteca puede ser utilizada simple e inmediatamente para realizar los diagramas utilizando la origen de datos interna con una estilización por defecto.

### 2.1. Creación de origen de datos interna

La origen de datos interna es un archivo XML contenido en una etiqueta HTML `<pre>`. Nada particular es necesario para utilizarlo, pero habrá que respetar la XSD especificada por el formato XML.

Con los datos presentados en forma de un XML, un objeto JavaScript `InternalDataSource` deberá ser utilizado para leer su contenido.

Este objeto respeta la interfaz `IdataSource` que comprende 2 métodos: `loadData(callback)` y `getDataMatrix()`. El método `loadData` comprende una función que será utilizada cuando los datos hayan sido cargados. El método `getDataMatrix` permite acceder a una vista abstracta de los datos, utilizable durante el resto de vida del programa.

### 2.2. Creación de un diagrama

Cuatro clases de diagramas podemos utilizar: `HistoDiagram`, `Histo3DDiagram`, `PieDiagram` y `LineDiagram`. La creación de un diagrama tras la carga de una base de datos interna se hace introduciendo en el constructor de la clase deseada el objeto DOM correspondiente en la casilla en la que debe ser colocada, así como el tipo de tratamiento deseado (fila o columna). Después, una vez que el objeto diagrama ha sido creado, podemos utilizar el método `setData(dataMatrix)` de el interfaz `IDiagram` para dar al diagrama una matriz de los datos.

El método `setData` rediseña inmediatamente el diagrama.

### 2.3. Ejemplo

Consideramos que la etiqueta `<pre>` que contiene el fichero XML tiene por identificador `pre1` y que la etiqueta `<canvas>` que contiene la zona de colocación tiene por identificador `canvas1`.

```
var ids = new InternalDataSource('pre1');
ids.loadData(function() {
// En cuanto la base de datos se carga la siguiente función se ejecuta
    var diag = new PieDiagram(document.getElementsByTagName('canvas1')[0], 'row');
    diag.setData(ids.getDataMatrix());
});
```

## Capítulo 3

# Utilización avanzada de la biblioteca

Presentamos aquí una utilización más compleja de la biblioteca, que permite utilizar archivos XML en un servidor externo, utilizar formatos diferentes al XML, y cambiar el diseño por defecto de los diagramas.

### 3.1. Utilización de origen de datos externas

Los orígenes de datos externas permiten que el archivo XML sea proporcionado por un tercero, que deberá no obstante respetar nuestro XSD. La utilización es similar a la de la base interna, con la diferencia de que cuando creemos el objeto `ExternalDataSource` deberemos pasar la URL del archivo XML en lugar del identificador de la etiqueta `<pre>`.

### 3.2. Utilización de conectores

Los conectores son uno de los aspectos esenciales de la modularidad de la biblioteca. Estos elementos deben ser capaces de crear un diagrama sin importar el tipo de datos. Por ejemplo podemos crear un diagrama reemplazando el formato XML por el formato JSON. El usuario del editor crea un enlace proporcionando al constructor del objeto `ConnectorDataSource` la implementación personalizada de las dos funciones `loadData` y `getDataMatrix`.

Un ejemplo en que se utiliza JSON figura en el código base del proyecto.

### 3.3. Estilización

La estilización permite personalizar en detalle la apariencia de los diagramas. Para mantener la coherencia con la provisión de los datos la estilización viene dada por un fichero XML, sea web o sea un proveedor externo.

### 3.3.1. Descripción del XML de estilización

La estilización consiste en dar un conjunto de color diferente al dado por defecto así como los parámetros de colocación de la leyenda.

```
<style>
<colors>
<color>Nombre_del_color</color>
...
<colors>
<legend>
<x>abscisa de la esquina superior izquierda</x>
<y>ordenada de la esquina superior izquierda</y>
<w>la longitud del rectángulo</w>
<h>altura del rectángulo</h>
</legend>
</style>
```

### 3.3.2. Utilización de fuentes internas y externas de estilización

La utilización de fuentes de estilización es muy similar a la de las fuentes de datos: para empezar creamos un objeto de tipo `InternalStyleSource` o `ExternalStyleSource` dando respectivamente un identificador de etiqueta `<pre>` o una url de archivo XML. Los dos métodos disponibles para estos objetos que utilizan la interfaz `IstyleSource` son `loadData` y `getStyleSource` que se comportan exactamente igual que su equivalente `IDataSource`.