

Google Chart

Con PHP



Francisco Arce
www.pacoarce.com

Google Chart

1. Bases de Google Chart
2. Conectar Google Chart con PHP y MySQL
3. Crear varias gráficas en una página
4. Leer un archivo plano para graficarlo
5. Leer un archivo JSON para graficarlo
6. Leer un archivo XML para graficarlo
7. El objeto DataTable

Google Chart

Con PHP



Francisco Arce
www.pacoarce.com

Google Chart

1. Cargar la librería
2. Preparar la información
3. Personalización de la gráfica
4. Dibujar la gráfica

<https://developers.google.com/chart/>

*Francisco Arce
www.pacoarce.com*

Hacer varias gráficas

Con PHP



Francisco Arce
www.pacoarce.com

Google Chart

Para cargar los datos necesitamos crear una tabla de dos dimensiones con el objeto:
`google.visualization.DataTable`

Google Chart

Un ***DataTable*** es una tabla bidimensional con filas y columnas, donde cada columna tiene un tipo de datos, una ID opcional y una etiqueta opcional.

Google Chart

Para dibujar más de una gráfica, necesitamos hacer una función de ***callback*** por gráfica, así como una etiqueta correspondiente.

Leer un archivo

Con PHP



Francisco Arce
www.pacoarce.com

Google Chart

```
$archivo = 'europa.txt';
// Leer las líneas del fichero
$renglones = file($archivo);
$data = array();
//
foreach($renglones as $renglon){
    $data[] = explode(';',trim($renglon));
}
```

Leer un archivo JSON

Con PHP



Francisco Arce
www.pacoarce.com

Google Chart

```
$archivo = 'europa.txt';
// Leer las líneas del fichero
$renglones = file($archivo);
$data = array();
//
foreach($renglones as $renglon){
    $data[] = explode(';',trim($renglon));
}
```

Leer un archivo XML

Con PHP



Francisco Arce
www.pacoarce.com

Google Chart

```
$xml = XMLReader::open($archivo);
while ($xml->read()) {
    if ($xml->nodeType== XMLReader::ELEMENT ) {
        $navegador = $xml->getAttribute("nombre");
        $porcien = $xml->getAttribute("porcien");
        print "[".$navegador.", ".$porcien."],";
    }
}
```

DataTable y DataView

Con PHP



Francisco Arce
www.pacoarce.com

DataTable y DataView

- Las gráficas son alimentadas por un tabla.
- **DataTable** y **DataView** son los objetos para “poblar” una gráfica.

DataTable y DataView

- Por medio del objeto ***DataTable*** creamos los datos para alimentar la gráfica.
- Con ***DataView*** podemos mostrar, ocultar u ordenar columnas y renglones de un objeto ***DataTable*** sin modificar los datos originales.

DataTable

Con PHP



Francisco Arce
www.pacoarce.com

DataTable

- Podemos crear y poblar una tabla de gráfica (dataTable) de cuatro formas:
 - Por el constructor DataTable y la poblamos por medio de los métodos ***addColumn()***, ***addRow()***, ***addRows()*** y ***setCell()***.

DataTable

- Podemos crear y poblar una tabla de gráfica (dataTable) de cuatro formas:
 - Por el constructor arrayToDataTable()

DataTable

- Podemos crear y poblar una tabla de gráfica (dataTable) de cuatro formas:
 - Un objeto de iniciación de JavaScript

DataTable

- Podemos crear y poblar una tabla de gráfica (dataTable) de cuatro formas:
 - Enviando un query Datasource()

SetColumns

Con PHP



Francisco Arce
www.pacoarce.com

SetColumns

- Especifica qué columnas son visibles en la vista.
- Las columnas no especificadas se ocultarán.
- Se trata de una matriz de índices de columna en la tabla / vista subyacente, o columnas calculadas.

SetColumns

- Si no llama a este método, el valor predeterminado es mostrar todas las columnas.
- La matriz también puede contener duplicados, para mostrar la misma columna varias veces.
- Las columnas se mostrarán en el orden especificado.

Francisco Arce

www.pacoarce.com

SetColumns

- ***columnIndexes*** - Una matriz de números y / o objetos (se puede mezclar):
- Los números especifican el índice de la columna de datos de origen que se va a incluir en la vista.
- Los datos se transmiten sin modificaciones.

SetColumns

- Si necesita definir explícitamente una función o propiedades de columna adicionales, especifique un objeto con una propiedad ***sourceColumn***.
- Los objetos especifican una columna calculada. Una columna calculada crea un valor sobre la marcha para cada fila y lo añade a la vista.

Francisco Arce

www.pacoarce.com

SetColumns

- El objeto debe tener las siguientes propiedades:
 - calc [función]: función que se llamará para cada fila de la columna para calcular un valor para esa celda. La función ***signature*** es func (dataTable, row), donde ***dataTable*** es la fuente DataTable, y ***row*** es el índice de la fila de datos de origen.

SetColumns

- La función debe devolver un único valor del tipo especificado por tipo.
- type [string] - El tipo de JavaScript del valor que devuelve la función calc.
- label [Opcional, cadena] - Una etiqueta opcional para asignar a esta columna generada.

SetColumns

- Si no se especifica, la columna de vista no tendrá ninguna etiqueta.
- *id* [Opcional, cadena]: ID opcional que se asigna a esta columna generada.
- ***sourceColumn*** - [Opcional, número] La columna de origen para usar como un valor; Si se especifica, no especifique la ***calc*** o la propiedad ***type***.

SetColumns

- Esto es similar a pasar un número en lugar de un objeto, pero le permite especificar una función y propiedades para la nueva columna.

SetColumns

- ***properties*** [Opcional, objeto] - Objeto que contiene propiedades arbitrarias para asignar a esta columna. Si no se especifica, la columna de vista no tendrá propiedades.

SetColumns

- [Opcional, cadena]: función que se asigna a esta columna. Si no se especifica, la función existente no se importará.