

Esercitazione Highcharts

Esercizio 1

Visualizzare i risultati della Web API popolazione.php attraverso un grafico a barre.

Soluzione

Per poter rappresentare i risultati della Web API popolazione attraverso un grafico a barre, utilizziamo la libreria javascript Highcharts¹. Innanzi tutto, occorre includere questa libreria nel file index.html. Nell'intestazione del file html, dopo aver incluso la libreria JQuery e prima di includere il nostro script js, includiamo la libreria di Highcharts:

```
<script src="http://code.highcharts.com/highcharts.js"></script>
```

A questo punto, nel nostro file index.html dobbiamo prevedere un div in cui inseriremo il grafico:

```
<div id="grafico_barre"></div>
```

Passiamo ora alla gestione della libreria Highcharts. Per poter creare un grafico Highcharts, basta chiamare nel punto di interesse del file js la seguente funzione:

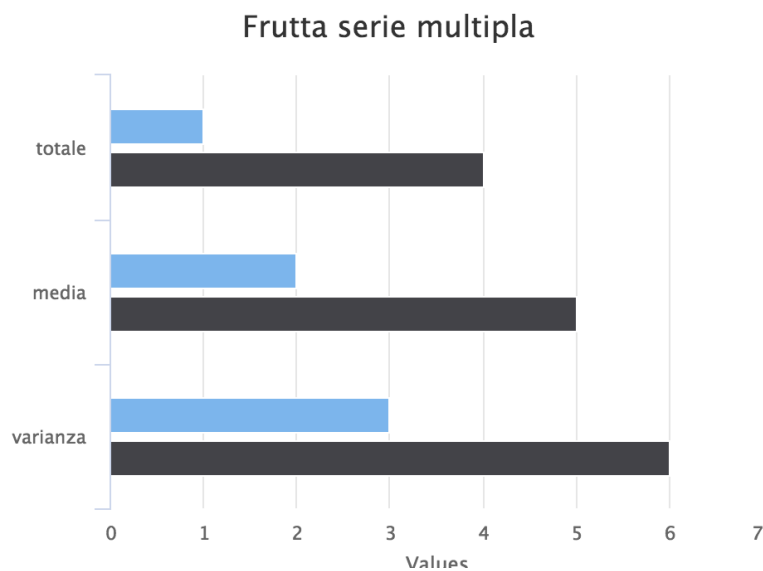
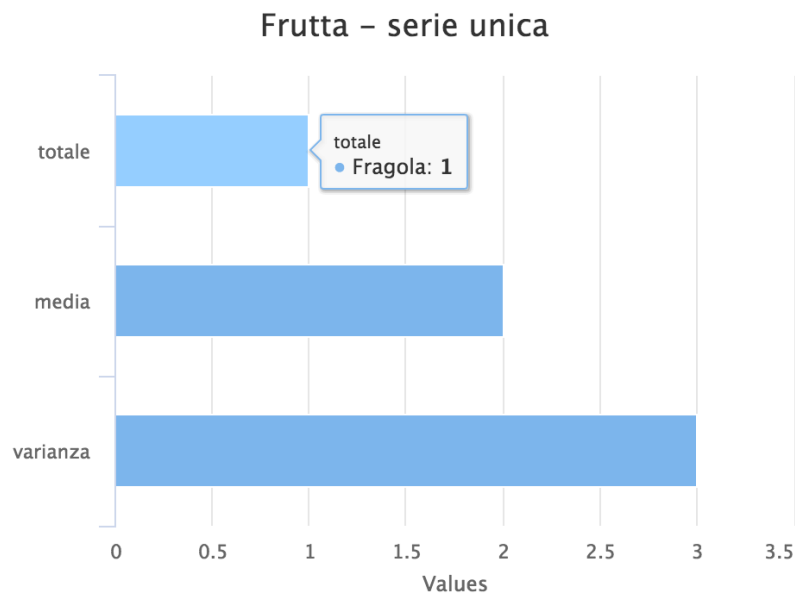
```
Highcharts.chart(nome_contenitore_nell_html, opzioni);
```

Il parametro opzioni contiene tutta una serie di opzioni da passare al grafico, tra cui il tipo di grafico e i dati da visualizzare. La variabile opzioni è un array associativo, che contiene coppie chiave-valore. Ad esempio:

```
var opzioni =  
{  
    chart : {type: 'bar'}, // tipo di grafico  
    series : dati // dati da visualizzare  
    ...  
}
```

Sofferamoci sulla variabile series. Questa variabile è un array numerico che contiene array associativi, uno per ogni serie che si vuole rappresentare. La figura sottostante mostra un esempio di grafico con una serie, mentre la successiva un esempio di grafico con due serie.

¹ <https://www.highcharts.com>



Dal punto di vista di Highcharts non ha molta importanza se si ha a che fare con una o più serie, è sufficiente creare un array associativo per ogni serie da rappresentare. Ogni serie è rappresentata da un array associativo in cui è obbligatorio specificare la chiave *data*, a cui corrisponde un array numerico contenente tutti i dati della serie, e una chiave opzionale *name*, che specifica il nome della serie:

```
Highcharts.chart('grafico_barre', {
  series : [{data : [1,2,3], name : ['Frugola']}, {data : [4,5,6], name : ['Banana']}],
});
```

Possiamo completare il nostro grafico highcharts, aggiungendo altre informazioni, come il titolo del grafico e le etichette da associare all'asse x.

```
Highcharts.chart('grafico_barre', {
  chart : {type: 'bar'},
  series : [{data : [1,2,3], name : ['Frugola']}, {data : [4,5,6], name : ['Banana']}],
  xAxis: {categories : ['totale', 'media', 'varianza']},
});
```

```
        title : {text: 'Frutta'}
    });
```

Ora possiamo posizionare il nostro grafico all'interno del nostro file main.js. Definiamo una funzione apposita che chiamiamo crea_grafico:

```
function crea_grafico(data)
{
    Highcharts.chart('grafico_barre', {
        chart : {type: 'bar'},
        series : [{data : [1,2,3], name : ['Fragola']}, {data :
[4,5,6], name : ['Banana']}],
        xAxis: {categories : ['totale', 'media', 'varianza']},
        title : {text: 'Frutta'}
    });
}
```

Invochiamo la funzione come callback della \$.getJSON relativa alla popolazione (va sostituita alla crea_tabella):

```
$.getJSON('api/popolazione.php', crea_grafico);
```

Verifichiamo da browser il corretto funzionamento dell'applicazione. Per ora il grafico dovrebbe mostrare la frutta. Il prossimo passo consiste nel modificare l'API dal lato PHP in modo da fornire al grafico il formato di dati corretto. Abbiamo detto che il formato deve essere:

```
[[{data: [lista-numerica], name: 'nome'},{data: [lista-numerica], name:
'nome'} ]
```

Ora, nel nostro caso, attualmente l'API popolazione.php restituisce per ogni provincia la popolazione. Distinguiamo due casi, uno in cui vogliamo una serie unica e l'altro in cui vogliamo la serie multipla. Aggiungiamo pertanto un parametro di ingresso all'API popolazione.php, che legge il tipo di serie. All'inizio del file popolazione.php scriviamo:

```
$serie = isset($_GET['serie']) ? $_GET['serie'] : 'singola';
```

Abbiamo associato un valore di default alla variabile \$serie. Assumiamo che \$serie possa assumere i valori 'singola' o 'multipla' e passiamo la variabile \$serie alla funzione get_risultato_grafico, che andremo a definire nel file funzioni.php:

```
$risultato = get_risultato_grafico($sql, $serie);
```

Il codice completo dell'API popolazione.php è il seguente:

```
<?php
header('Content-Type: application/json');
include('../php/funzioni.php');

$serie = isset($_GET['serie']) ? $_GET['serie'] : 'singola';
```

```

$select = array('Provincia', 'SUM(Popolazione) AS Popolazione');
$from = "ComuniTerremotati";
$where = null;
$optional = "GROUP BY Provincia";
$sql = costruisci_query($select,$from,$where,$optional);
$resultato = get_resultato_grafico($sql, $serie);
echo $resultato;

```

?>

Ora apriamo il file funzioni.php e definiamo una nuova funzione, in cui copiamo parte del codice della funzione get_resultati

```

function get_resultato_grafico($sql, $serie)
{
    include('../api/config.php');
    $resultato = select($db,$sql);
    for($i = 0; $i < count($resultato); $i++)
    {
        foreach($resultato[$i] as $chiave => $valore)
        {
            // codice da inserire
        }
    }
    closeDB($db);
    return json_encode($resultato);
}

```

Dobbiamo distinguere i due casi, serie multipla e serie singola:

```

if($serie == 'singola')
    // codice
else
    // codice

```

Nel caso di serie singola, dobbiamo produrre un array series fatto nel seguente modo:

```

[{"data": [{"popolazione1", "popolazione2"}, "name": "Popolazione"}]

```

Inoltre va aggiunto un array categories che contiene i nomi delle province:

```

{"categories": [provincia1, provincia2]}

```

Possiamo quindi creare un array associativo di questo tipo:

```

{
    series:      [{"data":      [{"popolazione1",      popolazione2},      name:
'Popolazione'}]},
    categories: [provincia1, provincia2]
}

```

Dal punto di vista del PHP, creiamo un array \$return e poi creiamo subito gli array che ci interessano:

```
$return = array();
$return['series'] = array();
$return['series'][0] = array();
$return['series'][0]['data'] = array();
$return['series'][0]['name'] = 'Popolazione';
$return['categories'] = array();
```

Ora scorriamo l'array \$risultato:

```
for($i = 0; $i < count($risultato); $i++)
{
    foreach($risultato[$i] as $chiave => $valore)
    {
        switch($chiave)
        {
            case 'Popolazione':
                $return['series'][0]['data'][] = intval($valore);
                break;
            case 'Provincia':
                $return['categories'][] = $valore;
                break;
        }
    }
}
```

Il codice ottenuto fino ad ora della `get_risultato_grafico` è il seguente:

```
function get_risultato_grafico($sql, $serie)
{
    include('../api/config.php');
    $risultato = select($db,$sql);
    $return = array();
    $return['series'] = array();
    $return['series'][0] = array();
    $return['series'][0]['data'] = array();
    $return['series'][0]['name'] = 'Popolazione';

    $return['xAxis'] = array();
    $return['xAxis']['categories'] = array();

    for($i = 0; $i < count($risultato); $i++)
    {
        foreach($risultato[$i] as $chiave => $valore)
        {
            switch($chiave)
            {
```

```

        case 'Popolazione':
            if($serie == 'singola')
                $return['series'][0]['data'][] =
intval($valore);
                break;
        case 'Provincia':
            if($serie == 'singola')
                $return['categories'][] = $valore;
            break;
        }
    }
}
closeDB($db);
return json_encode($return);
}

```

Passiamo ora al lato javascript. Possiamo sostituire la nostra serie di frutta con i risultati della nostra API come segue (ricordiamoci di chiamare l'API popolazione specificando il campo serie='singola'):

```

function crea_grafico(data)
{
    var opzioni = {
        chart: {type      : 'bar'},
        title : {text: 'Popolazioni prima del sisma'},
        series: data.series,
        xAxis: {categories: data.categories }
    };
    Highcharts.chart('grafico_barre', opzioni);
}

```

La chiamata alla funzione crea_grafico è la seguente:

```
$.getJSON('api/popolazione.php?serie=singola', crea_grafico);
```

Passiamo ora al caso di serie multipla. In questo caso dobbiamo produrre un array series di questo tipo:

```

[{'data' : [popolazione1], name : 'provincia1'},{'data' : [popolazione2],
name : 'provincia2'}]

```

Inoltre possiamo aggiungere un array categories contenente l'unica categoria:

```
{'categories' : 'Popolazione'}
```

L'array restituito dall'API dovrebbe essere fatto nel seguente modo:

```

{
    series: [{'data' : [popolazione1], name : 'provincia1'},{'data' :
[popolazione2], name : 'provincia2'}],

```

```

        categories : 'Popolazione'
    }

```

Il codice PHP per ottenere l'array \$return diventa il seguente:

```

$return['categories'] = array('Popolazione');
for($i = 0; $i < count($risultato); $i++)
{
    $return['series'][$i] = array();
    foreach($risultato[$i] as $chiave => $valore)
    {
        switch($chiave)
        {
            case 'Popolazione':
                $return['series'][$i]['data'] =
array(intval($valore));
                break;
            case 'Popolazione':
                $return['series'][$i]['name'] = $valore;
                break;
        }
    }
}

```

Il codice completo della crea_risultato_grafico è il seguente:

```

function get_risultato_grafico($sql, $serie)
{
    include('../api/config.php');
    $risultato = select($db,$sql);
    $return = array();
$return['series'] = array();
if($serie == 'singola')
{
    $return['series'][0] = array();
    $return['series'][0]['data'] = array();
    $return['series'][0]['name'] = 'Popolazione';
    $return['categories'] = array();
}
else
$return['categories'] = array("Popolazione");
for($i = 0; $i < count($risultato); $i++)
{
    if($serie == 'multipla')
        $return['series'][$i] = array();
    foreach($risultato[$i] as $chiave => $valore)
    {
        switch($chiave)
        {
            case 'Popolazione':

```

```

        if($serie == 'singola')
            $return['series'][0]['data'][] =
intval($valore);
        else
            $return['series'][$i]['data'] =
array(intval($valore));
        break;
    case 'Provincia':
        if($serie == 'singola')
            $return['categories'][] = $valore;
        else
            $return['series'][$i]['name'] = $valore;
        break;
    }
}
}
closeDB($db);
return json_encode($return);
}

```

Dal punto di vista del javascript il codice è lo stesso, occorre cambiare solo la chiamata all'API:

```
$.getJSON('api/popolazione.php?serie=singola', crea_grafico);
```

Se si vuole cambiare tipo di grafico, è sufficiente cambiare il parametro `chart.type` nelle opzioni di Highcharts e sistemare qualche altro parametro.

Esercizio 2

Creare un grafico a torta, usando il tipo 'pie'.