Chapitre 19
Graphiques de gestion

1. Différentes solutions de conception de graphiques de gestion

Il existe de multiples solutions permettant d'élaborer des graphiques de gestion (histogrammes, représentations en secteurs, nuages de points...) disponibles sur des pages Web.

Les outils à disposition sont souvent des librairies écrites en PHP ou encore en Java. Citons par exemple :

- JpGraph : http://jpgraph.net/
- JFreeChart : http://www.jfree.org/jfreechart/
- GraphoViz : http://www.graphviz.org/
- Highcharts JS : http://www.highcharts.com/

Google a développé de son côté une série d'API permettant de concevoir simplement des graphiques de gestion de bonne facture. Ces API sont regroupées sous l'appellation Google Charts.

2. Exemples d'utilisation des API Google Charts

Il ne peut être ici question de reproduire l'intégralité des exemples proposés par Google pour illustrer les très nombreuses possibilités offertes par ces API.

Nous nous contenterons ici de proposer quelques exemples significatifs.

Remarque

Les exemples fournis par Google sont disponibles à l'adresse suivante : https://developers.google.com/chart/?hl=fr

2.1 Exemple 1 : Tracé d'un histogramme

L'objectif de ce graphe est simple, assurer la présentation sous forme d'un histogramme (barres verticales) de la production d'un produit LAMBDA (exprimée en tonnes) de trois pays (Allemagne, France et Italie) et ceci pour les années 2010, 2011 et 2012.

Une représentation en "ligne brisée" de la moyenne de la production par année sera superposée sur l'histogramme.

Section HTML <body>

Le code placé dans cette section sera quasiment identique pour l'ensemble des exemples de ce chapitre. Il est ici reproduit intégralement :

```
<!-- Début section body du script HTML -->
<body>

    <!-- Titre du traitement -->
    <h1>Editions ENI - JavaScript - COMBO</h1>

    <!-- Début script JavaScript -->
    <script type="text/javascript">

            /* Affichage du nom du script */
            alert("COMBO");

    </script>

    <!-- Définition de la division d'affichage du graphique -->
```

```
    <div id="chart_div" style="width: 900px; height: 500px;"></div>

    <!-- Affichage du code source -->
    <br /><br /><br />
    <center>
    <a href="JavaScript:window.location='view-source:' +
    window.location">
            Code source
    </a>
    </center>

</body>
```

Une division (`chart_div`) est prévue dans ce code pour l'affichage du graphique (par l'intermédiaire de la fonction `tracerGraphe`, intégrée à la section `<head>`) :

```
<!-- Définition de la division d'affichage du graphique -->
<div id="chart_div" style="width: 900px; height: 500px;"></div>
```

Section HTML <head>

Comme pour la section précédente, le code source de cette section est entièrement listé ci-après :

```
<!-- Début en-tête script HTML -->
<head>

    <!-- Balise meta -->
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    <!-- Titre du script HTML -->
    <title>COMBO</title>

    <!-- Chargement de l'API Google Chart -->
    <script type="text/javascript"
    src="https://www.google.com/jsapi"></script>

    <!-- Chargement du module visualization en version 1
    avec en option le package corechart -->
    <script type="text/javascript">
        google.load('visualization', '1', {packages: ['corechart']});
    </script>

    <!-- Définition de l'histogramme -->
    <script type="text/javascript">
```

```
        /* Fonction tracerGraphe */
        function tracerGraphe()
        {

            /* Données à représenter graphiquement */
            var donnees = google.visualization.arrayToDataTable([
                ['Année', 'Allemagne', 'France', 'Italie', 'Moyenne'],
                ['2010', 100, 110, 150, 120],
                ['2011', 120, 130, 200, 150],
                ['2012', 140, 120, 220, 160],
            ]);

            /* Options de représentation graphique */
            /* NB1 : La 4ème série (numérotation à partir de 0 des
                     séries) est une moyenne représentée par une ligne */
            /* NB2 : Pour les nombreuses options veuillez consulter
            /*       https://developers.google.com/chart/interactive/
                     docs/gallery/combochart */
            var options =
            {
                title : 'Production annuelle par pays',
                vAxis: {title: "Tonnes"},
                hAxis: {title: "Années"},
                seriesType: "bars",
                series: {3: {type: "line"}}
            };

            /* Instanciation du graphique */
            var chart = new google.visualization.ComboChart(document
            .getElementById('chart_div'));

            /* Dessin du graphique */
            chart.draw(donnees, options);

        }

        /* Affichage automatique du graphique */
        google.setOnLoadCallback(tracerGraphe);

    </script>

</head>
```

La première séquence spécifique dans cet exemple est le chargement de l'API Google Chart :

```
<!-- Chargement de l'API Google Chart -->
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>
```

Suit le chargement du module `visualization` (en version 1) avec le package dédié aux graphiques basiques, `corechart` :

```
<!-- Chargement du module visualization en version 1
avec en option le package corechart -->
<script type="text/javascript">
    google.load('visualization', '1', {packages: ['corechart']});
</script>
```

Passons maintenant à l'étude de l'unique fonction (`tracerGraphe`) assurant le tracé du graphique.

La fonction débute par le stockage des données à représenter graphiquement dans un tableau mémoire :

```
/* Données à représenter graphiquement */
var donnees = google.visualization.arrayToDataTable([
    Année', 'Allemagne', 'France', 'Italie', 'Moyenne'],
    ['2010', 100, 110, 150, 120],
    ['2011', 120, 130, 200, 150],
    ['2012', 140, 120, 220, 160],
]);
```

Ce tableau est sans surprise, la première ligne indique la structure (`Année` et les trois pays), les lignes suivantes étant des lignes de données.

Les noms des pays et les mentions des années serviront tout naturellement de légendes.

Des options sont disponibles pour affiner la présentation du graphique. Vous pourrez trouver le détail de ces options à l'adresse Internet mentionnée dans le script.

Les options retenues sont :

```
var options =
{
    title : 'Production annuelle du produit LAMBDA par pays',
    vAxis: {title: "Tonnes"},
    hAxis: {title: "Années"},
    seriesType: "bars",
    series: {3: {type: "line"}}
};
```

L'option `title` servira de titre sur le graphique, `vAxis` et `hAxis` seront les légendes des axes, `seriesType` indique qu'un graphique en barres (histogramme) est souhaité et enfin l'option `series` précise que la quatrième série (la moyenne) doit être affichée sous forme d'une "ligne brisée". Vous noterez que la numérotation des séries débute à zéro.

Remarque

Vous pourrez consulter les nombreuses options à l'adresse suivante : https://developers.google.com/chart/interactive/docs/gallery/combochart

Il reste ensuite à instancier le graphique et en assurer le tracé sur la base des données stockées dans le tableau `donnees` et des `options` :

```
/* Instanciation du graphique */
var chart = new google.visualization.ComboChart(document
.getElementById('chart_div'));

/* Dessin du graphique */
chart.draw(donnees, options);
```

Le tracé effectif est déclenché finalement par la directive suivante :

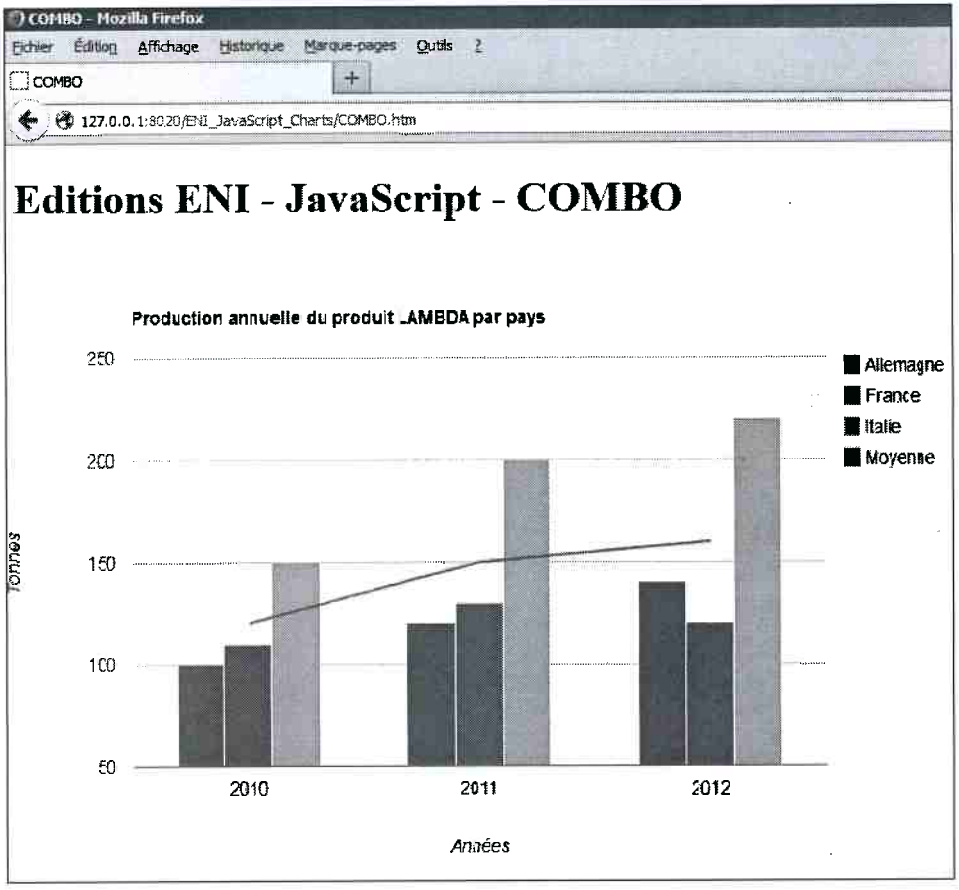
```
/* Affichage automatique du graphique */
google.setOnLoadCallback(tracerGraphe);
```

Remarque

La méthode `setOnLoadCallback` est une solution plus efficace que l'appel de la fonction `tracerGraphe` par un `<body onload='tracerGraphe()'>`.

Exécution du script

À l'exécution, le script `COMBO.htm` donne ceci :



2.2 Exemple 2 : Tracé d'un graphique en secteurs

La France vient de se qualifier péniblement pour la Coupe du Monde de football qui se jouera au Brésil en 2014. Elle fait partie du groupe E au même titre que la Suisse, l'Équateur et le Honduras. Le graphique en secteurs développé montre le pourcentage de chance, accordée par mes collègues de travail, de terminer premier de ce groupe pour chacune des équipes.

Section HTML <body>

Le code source de cette section ne présente aucune particularité et n'est pas reproduit ici.

Section HTML <head>

Cette section débute comme dans l'exemple précédent par le chargement de l'API Google Chart et le chargement du module `visualization` (avec le package corechart).

Attardons-nous maintenant sur le code de la fonction `tracerGraphe` :

```
/* Fonction tracerGraphe */
function tracerGraphe()
{

    /* Données à représenter graphiquement */
    var donnees = google.visualization.arrayToDataTable([
        ['Pays', 'Qualification (%)'],
        ['Suisse', 30],
        ['Equateur', 20],
        ['France', 40],
        ['Honduras', 10]
    ]);

    /* Options de représentation graphique */
    /* NB : Pour les nombreuses options veuillez consulter
    /*      https://developers.google.com/chart/interactive/
           docs/gallery/piechart */
    var options =
    {
        title: 'Pourcentage de chance pour être 1er du groupe E de la
        Coupe du Monde de Football 2014',
        colors:['Red', 'Green', 'Blue', 'Grey'],
```

```
        slices: {
            0: {offset: 0.01},
            1: {offset: 0.01},
            2: {offset: 0.20},
            3: {offset: 0.01}
        }
    };

    /* Instanciation du graphique */
    var chart = new google.visualization.PieChart(document
    .getElementById('chart_div'));

    /* Tracé du graphique */
    chart.draw(donnees, options);

};
```

Les données à présenter graphiquement sont stockées dans un tableau nommé `donnees`. Le premier champ constitue le nom du pays et le second le pourcentage de chance de terminer premier.

Viennent ensuite les options d'affichage.

Dans notre cas, il est prévu de choisir la couleur (`colors`) pour chacun des secteurs, le rouge pour la Suisse, le vert pour l'Équateur...

Le paramètre `slices` permet de ressortir les différents secteurs du graphe. À titre d'exemple, la France, grande favorite du groupe, a un secteur avec une valeur d'offset plus importante que ses challengers.

Remarque

Vous pourrez consulter les nombreuses options à l'adresse suivante : https://developers.google.com/chart/interactive/docs/gallery/piechart

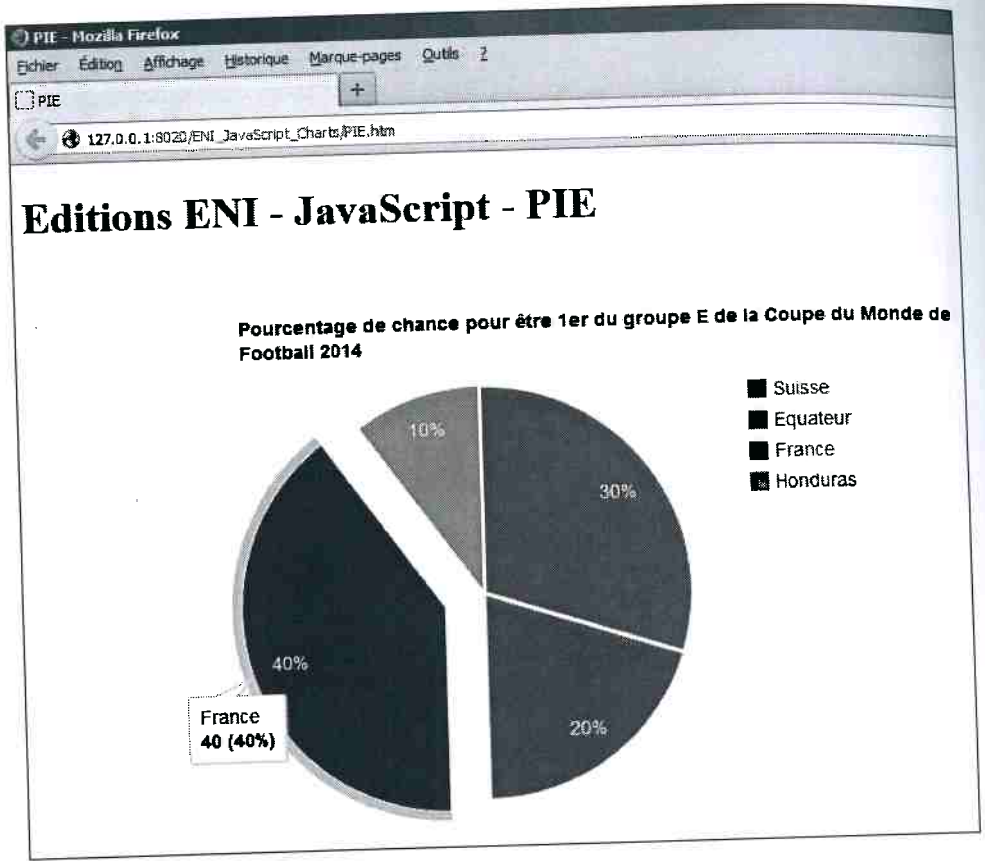
La fonction se termine comme dans l'exemple 1 par l'instanciation et le tracé du graphique.

Le déclenchement de l'affichage du graphique dans la division HTML `chart_div` se fait aussi comme dans le précédent exemple après la fin de la fonction `tracerGraphe`, comme suit :

```
/* Affichage automatique du graphique */
google.setOnLoadCallback(tracerGraphe);
```

Exécution du script

À l'exécution, le script `PIE.htm` donne ceci :



2.3 Exemple 3 : Tracé d'une carte

Depuis l'exemple précédent, vous savez que la France vient de se qualifier péniblement pour la Coupe du Monde de football qui se jouera au Brésil en 2014. Elle a été placée par tirage dans le groupe E avec la Suisse, l'Équateur et le Honduras.

L'objectif de cet exemple est de représenter la position géographique de ces quatre pays sur une carte mondiale.

Section HTML <body>

Une fois de plus, le code source de cette section ne présente aucune particularité.

Section HTML <head>

Cette section débute comme dans l'exemple précédent par le chargement de l'API Google Chart et le chargement du module `visualization`. Il y a quand même une petite nuance, le remplacement du package `corechart` par le package `geochart`.

```
<!-- Chargement du module visualization en version 1
avec en option le package geochart -->
<script type="text/javascript">
    google.load('visualization', '1', {packages: ['geochart']});
</script>
```

Dans la fonction `tracerGraphe`, vous trouverez le stockage des données à représenter graphiquement dans un tableau :

```
/* Données à représenter graphiquement (Groupe E) */
var donnees = google.visualization.arrayToDataTable([
    ['Pays', 'Indicateur'],
    ['Switzerland', 500],
    ['Ecuador', 500],
    ['France', 500],
    ['Honduras', 500]
]);
```

La première colonne représente le nom des pays. Vous noterez qu'il est obligatoire de nommer les pays sous leur appellation internationale (ISO), donc Switzerland au lieu de Suisse, afin que le pays soit bien repéré sur la carte.

La valeur annoncée dans la deuxième colonne (nommée ici `Indicateur`) définit la densité de la couleur représentant chaque pays. L'option a été prise de choisir une valeur commune (500) afin que les quatre pays soient représentés dans le même vert (la couleur par défaut). Jouer sur la valeur de cette deuxième colonne aurait par exemple été intéressant s'il avait été souhaité comparer la population de chaque pays.

Viennent ensuite les habituelles options :

```
/* Options de représentation graphique */
/* NB : Pour les nombreuses options, veuillez consulter
/*      https://developers.google.com/chart/interactive/docs/
        gallery/geochart */
var options =
{
    region: 'world',
    backgroundColor: 'LightBlue',
    fill: 'Green',
    datalessRegionColor: 'Yellow',
    legend: 'null'
};
```

Le paramètre `region` sert à indiquer dans notre cas qu'une carte du monde est souhaitée (il est aussi possible de retenir un affichage par continent, par pays...).

La couleur de fond (`backgroundColor`) de la carte est `LightBlue`. Cela colore logiquement les mers et océans en bleu clair.

L'option `fill` donne la couleur retenue (le vert) pour identifier les pays sur la carte. Les autres pays apparaîtront en jaune (option `datalessRegion`).

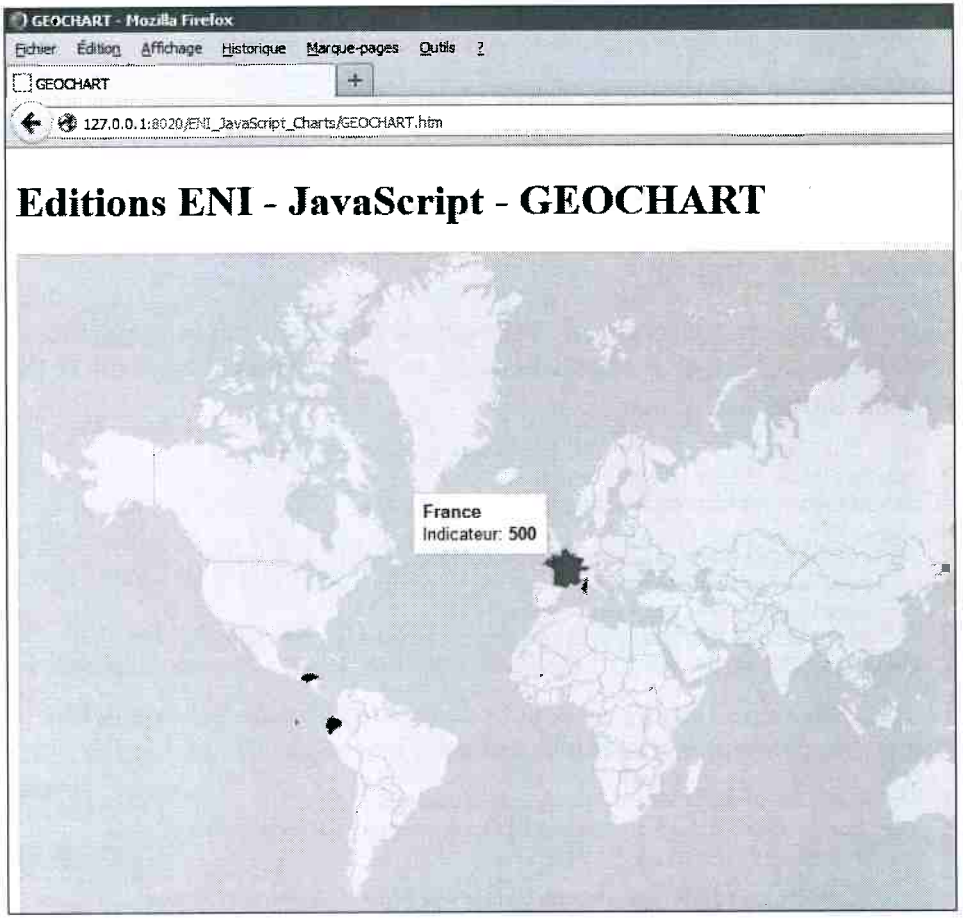
Enfin, il n'est pas prévu de faire figurer une légende sur la carte.

Remarque

Vous pourrez consulter les nombreuses options à l'adresse suivante : https://developers.google.com/chart/interactive/docs/gallery/geochart

Exécution du script

Le script GEOCHART.htm génère cet affichage :



2.4 Exemple 4 : Tracé d'une jauge

La présentation de phénomènes, d'indicateurs sous forme de jauges devient un grand classique. Dans cet exemple, il est prévu d'afficher la vitesse à l'instant t d'une voiture sur un graphique de type compteur de vitesse.

Section HTML <body>

Une fois de plus, le code source de cette section ne présente aucune particularité.

Section HTML <head>

Cette section débute comme dans l'exemple précédent par le chargement de l'API Google Chart et le chargement du module `visualization` (avec le package `gauge`).

```
<!-- Chargement de l'API Google Chart -->
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>

    <!-- Chargement du module visualization en version 1
    avec en option le package gauge -->
    <script type="text/javascript">
    google.load('visualization', '1', {packages: ['gauge']});
</script>
```

La fonction `tracerGraphe` débute comme d'habitude par le stockage des données à représenter graphiquement dans un tableau :

```
/* Données à représenter graphiquement */
var donnees = google.visualization.arrayToDataTable([
    ['Label', 'Value'],
    ['Km/h', 210]
]);
```

Le tableau comporte deux champs (colonnes), `Label` et `Value` et un seul enregistrement.

La mention `Km/h` apparaîtra sur le compteur de vitesse ainsi que la vitesse (210) bien entendu.

Les options d'affichage sont nombreuses. Dans notre cas nous avons retenu celles-ci :

```
/* Options de représentation graphique */
/* NB : Pour les nombreuses options, veuillez consulter
/*      https://developers.google.com/chart/interactive/
        docs/gallery/gauge */
var options =
{
    width: 500, height: 500,
    yellowFrom:160, yellowTo: 240,
    redFrom: 240, redTo: 320,
    majorTicks: [0, 40, 80, 120, 160, 200, 240, 280, 320],
    minorTicks: 2,
    min: 0,
    max: 320
};
```

Les paramètres `width` et `height` (exprimés en pixels) définissent la largeur et la hauteur de la jauge (compteur de vitesse).

Une partie du cadran du compteur comportera une zone jaune entre 160 (`yellowFrom`) et 240 (`yellowTo`) Km/h et une seconde partie une zone rouge entre 240 (`redFrom`) Km/h et 320 (`redTo`) Km/h.

Le compteur sera gradué (`majorTicks`) tous les 40 Km/h et ceci à partir de 40 Km/h. Une sous-graduation tous les 20 Km/h est aussi prévue (`minorTicks`).

Enfin la graduation du compteur s'échelonne de 0 Km/h (`min`) à 320 Km/h (`max`).

Remarque

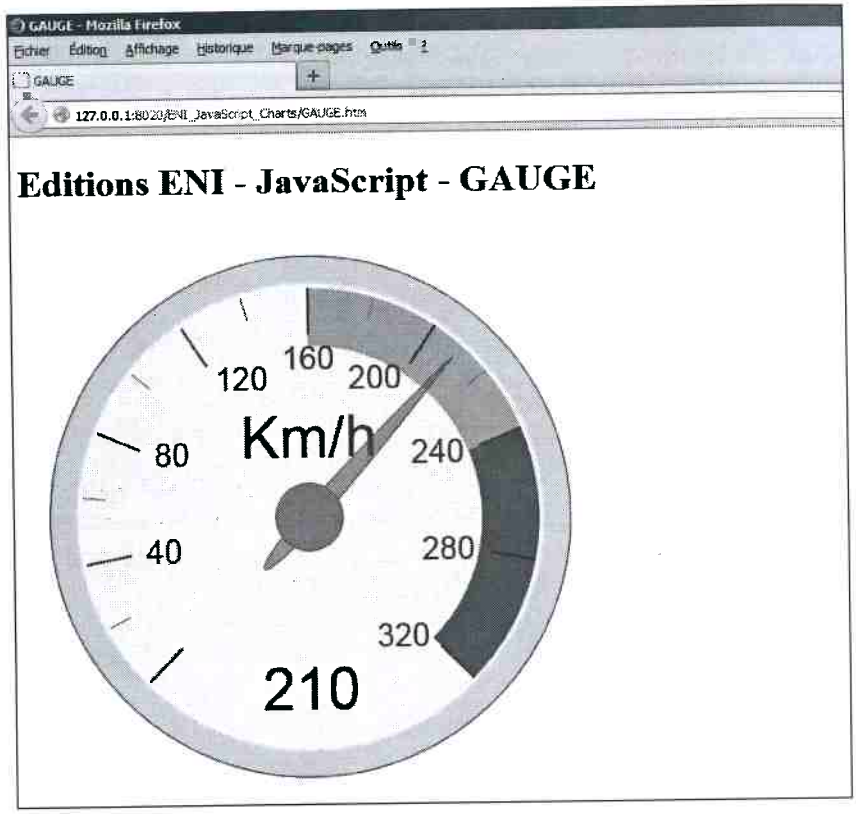
Vous pourrez consulter les nombreuses options à l'adresse suivante : https://developers.google.com/chart/interactive/docs/gallery/gauge

La fonction se termine comme dans l'exemple 1 par l'instanciation et le tracé du graphique.

Le déclenchement de l'affichage du graphique dans la division HTML `chart_div` se fait aussi comme dans les précédents exemples après la fin de la fonction `tracerGraphe`.

Exécution du script

À l'exécution, le script `GAUGE.htm` donne ceci :



2.5 Exemple 5 : Tracé d'un timeline

Le positionnement d'événements sur un axe de temps (*timeline*) est une présentation graphique très courante.

Fixons-nous comme objectif de représenter graphiquement sur un *timeline* les récents présidents de la République Française (François MITTERRAND, Jacques CHIRAC et Nicolas SARKOZY) avec mention de leur durée de mandat(s).

Section HTML <body>

Le code source de cette section ne présente aucune particularité par rapport aux exemples précédents.

Section HTML <head>

Comme dans l'exemple précédent, cette section débute par le chargement de l'API Google Chart et le chargement du module `visualization`. Le package `timeline` remplace le package `gauge`.

```
<!-- Chargement de l'API Google Chart -->
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>

<!-- Chargement du module visualization en version 1
avec en option le package timeline -->
<script type="text/javascript">
    google.load('visualization', '1', {packages: ['timeline']});
</script>
```

La définition des données à représenter graphiquement se fait en deux phases dans la fonction `tracerGraphe`.

Commençons par la description de la structure du tableau des données :

```
/* Définition de la structure du tableau à représenter graphiquement */
var dataTable = new google.visualization.DataTable();
dataTable.addColumn({
    type : 'string',
    id : 'Président de la République Française'
});
dataTable.addColumn({
```

```
    type : 'string',
    id : 'mandat'
});
dataTable.addColumn({
    type : 'date',
    id : 'debut_mandat'
});
dataTable.addColumn({
    type : 'date',
    id : 'fin_mandat'
});
```

Le premier champ de type `string` servira au stockage du prénom et du nom du Président de la République.

Le deuxième champ également de type `string` permettra le stockage de la durée du mandat (des mandats).

Les champs 3 et 4 servent à paramétrer la longueur de la barre horizontale correspondant à chaque président.

Viennent ensuite les lignes de données :

```
/* Lignes de données */
dataTable.addRows([
    ['François MITTERRAND', '(1981-1995)', new Date(1981, 5, 21),
    new Date(1995, 5, 17)],
    ['Jacques CHIRAC', '(1995-2007)', new Date(1995, 5, 17),
    new Date(2007, 5, 16)],
    ['Nicolas SARKOZY', '(2007-2012)', new Date(2007, 5, 16),
    new Date(2012, 5, 15)]
]);
```

Vous noterez le format particulier pour les dates de début et de fin de mandat (année, mois, jour).

Reste à voir les options :

```
/* Options de représentation graphique */
/* NB : Pour les nombreuses options veuillez consulter
/*      https://developers.google.com/chart/interactive/
        docs/gallery/timeline */
var options = {
    colors : ['Red', 'Blue', 'Orange']
};
```

L'option `colors` indique la couleur réservée à chaque président, rouge (`red`) pour François MITTERRAND par exemple. Les noms des couleurs peuvent être remplacés par les codes hexadécimaux associés.

Remarque

Vous pourrez consulter les nombreuses options à l'adresse suivante : https://developers.google.com/chart/interactive/docs/gallery/gauge

Comme dans l'ensemble des exemples de ce chapitre, la fonction `tracerGraphe` se termine par l'instanciation et le tracé du graphique.

Le déclenchement de l'affichage du graphique dans la division HTML `chart_div` se fait aussi après la fin de la fonction `tracerGraphe`.

Exécution du script

Le script `TIMELINE.htm` affiche ceci :