

TP n° 01 – JavaScript

Ressource R209

Ce TP va être l'occasion de s'initier à la programmation JavaScript coté « client » (sur le navigateur).

1 Affichages de courbes avec Chart.js :

Il existe une multitude de bibliothèques d'affichage de courbes en JavaScript sur Internet

(https://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks). Parmi elles se trouve `charts.js` (<https://github.com/chartjs/Chart.js>) qui se sert de la balise HTML5 `canvas`.

Pour l'utiliser, il suffit de charger la bibliothèque (`charts.min.js`) en début de page HTML puis, dans un script JavaScript spécifique, de paramétrer le graphique à réaliser et d'appliquer la configuration au contexte graphique de la balise `canvas` désirée.

Noter dans l'exemple ci-dessous, l'utilisation de l'objet `window` et de la redéfinition de la méthode `onload` pour exécuter le code à la fin du chargement de la page ainsi que la création dynamique de la propriété `graph1` afin de pouvoir modifier le graphique à posteriori.

```
var jours=[1,2,3,4,5,6,7,8,9,10];
var tmin=[7.0,7.0,4.4,2.7,8.2,2.3,4.1,5.5,11.6,8.2];
var courbe1={
  label: "Température minimum en °C",
  backgroundColor: "darkcyan",
  borderColor: "darkcyan",
  data: tmin,
  fill: false
};
var config = {
  type: "line",
  data: {labels: jours, datasets: [courbe1]},
  options: {title: {display: true, text: "Béziers en 2007"}}
};

window.onload=function() {
  var ctx=document.getElementById("canvas").getContext("2d");
  window.graph1=new Chart(ctx,config);
};
```

1. Tester la page `tp1-1.html`.
2. On veut maintenant pouvoir changer la couleur de la courbe en créant automatiquement les éléments d'une liste de sélection HTML à partir du tableau de couleurs JavaScript suivant :

```
var colors=["black","yellow","blue","magenta","aqua","coral","red","chartreuse"];
```

Ajouter dans la page HTML, la balise `select` avec l'identifiant `couleur` et une option vide.

Dans le JavaScript, faire une boucle sur les éléments du tableau trié (utiliser la méthode `sort()`) en créant des balises `option` avec l'attribut `value` égal à l'index de la couleur dans le tableau et le contenu HTML fixé au nom de la couleur, puis en l'ajoutant à la liste déroulante (voir `createElement()`, `setAttribute()`, `innerHTML` et `appendChild()`).

3. Enfin pour gérer les actions sur la liste déroulante, on peut utiliser un « `eventListener` » qui a pour avantage de d'avoir accès à l'objet `this` correspondant à la liste elle-même. On peut alors récupérer l'index sélectionné ainsi que la valeur du texte :

```
var selectCouleur=document.getElementById("couleur");
selectCouleur.addEventListener("change", function() {
  var val=this.options[this.selectedIndex].value;
  var txt=this.options[this.selectedIndex].text;
  console.log("Couleur "+this.selectedIndex+": "+val+" -> "+txt);
});
```

Tester le code ci-dessus et vérifier que l'on voit bien dans la console WEB, la couleur sélectionnée.

Changer alors la couleur de la courbe1 (voir propriétés `backgroundColor` et `borderColor`) et actualiser le graphique à l'aide de la méthode `update()` sur l'objet `window.graph1`.

4. On peut ajouter des descriptions sur les axes dans la partie options avec la propriété scales :

```
var axeX={
  display: true,
  scaleLabel: {display: true, labelString: "Numéro du jour"}
};
var axeY={
  display: true,
  scaleLabel: {display: true, labelString: "Température en °C"}
};
config.options.scales={xAxes: [axeX], yAxes: [axeY]};
```

Tester cette fonctionnalité.

5. Comment afficher ou non les échelles en utilisant une case à cocher (identifiant « *echelles* ») cochée par défaut (dans ce cas, il faut définir l'attribut `checked`) ? Utiliser la même technique qu'au § 1.3 pour gérer les changements de la case à cocher, sachant que `this.checked` reflétera l'état de la case (cochée ou non). Le code suivant permet alors de mettre à jour l'affichage :

```
config.options.scales.xAxes[0].scaleLabel.display=this.checked;
config.options.scales.yAxes[0].scaleLabel.display=this.checked;
window.graph1.update();
```

6. Ajouter ensuite une 2^{ème} courbe avec la couleur `darkorchid`, sur le même graphique à partir des relevés de vitesse du vent suivants :

```
var wind=[68.5,79.6,42.6,64.8,50.0,18.5,18.5,38.9,38.9,35.2];
```

Après avoir créé l'objet `courbe2` comme au début, on l'ajoute dynamiquement à la fin du tableau `datasets` avec la méthode `push()` :

```
config.data.datasets.push(courbe2);
```

7. Seulement on remarque que l'axe des Y n'indique pas la bonne unité ! Et que cela « tasse » un peu la courbe de température à cause des gammes de valeurs des 2 courbes (de 2 à 12 °C et de 20 à 80 Km/h). Il faut donc créer une seconde échelle sur la droite du graphique et indiquer des valeurs minimum et maximum différentes sur chaque axe.

On commence par modifier l'objet gérant les paramètres de l'axe des Y en ajoutant les 3 propriétés en gras ci-dessous. Puis on le duplique en adaptant les valeurs pour l'axe 2 (`axeY2`) à droite.

```
var axeY1={
  id: "yTmin",
  position: "left",
  ticks: {suggestedMin: 2, suggestedMax: 12},
  display: true,
  scaleLabel: {display: true, labelString: "Température en °C"}
};
```

Ensuite on indique l'identifiant à utiliser pour chaque courbe dans la propriété `yAxisID`.

```
courbe1.yAxisID="yTmin";
courbe2.yAxisID="yWind";
```

Puis on ajoute les 2 échelles dans les options du graphique :

```
config.options.scales.yAxes=[axeY1, axeY2];
```

NB : On peut désactiver l'affichage des lignes horizontales d'une échelle en ajoutant la propriété suivante sur l'axe en question :

```
gridLines: {drawOnChartArea: false}
```

8. On peut obtenir les valeurs numériques des 2 courbes dans la même « boîte » quand on passe la souris sur le graphique en ajoutant la propriété suivante dans les options :

```
tooltips: {mode: "index", intersect: false}
```

9. Comment modifier la couleur de n'importe quelle courbe ? On pourra créer dynamiquement des cases « radio » permettant de sélectionner le numéro de courbe à colorier (Créer la variable globale `idCourbe` fixée à 0 par défaut). On adaptera alors le code du § 1.3 pour changer la couleur de l'objet `config.data.datasets[idCourbe]` plutôt que `courbe1`.

On pourra utiliser la méthode `forEach()` sur le tableau `config.data.datasets` pour déterminer le nombre courbes existantes...

10. Comment intégrer un plugin de zoom au graphique ?