



Les éléments graphiques

Le langage HTML 5

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS



Séquence 4 : Les éléments graphiques

Objectifs spécifiques : A la suite de cette séquence, l'étudiant doit être capable de :

1. Décrire les nouveaux éléments graphiques de HTML 5
2. Dessiner dans une page HTML des graphiques à la volée en utilisant JavaScript
3. Dessiner dans une page HTML des graphiques vectoriels
4. Afficher des cartes dans une page web

Séquence 4 : Les éléments graphiques

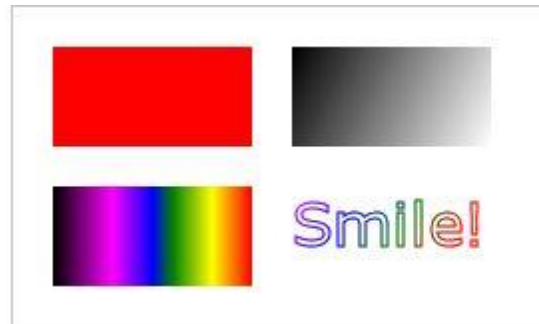
Plan de la séquence :

1. Introduction
2. Dessin de graphiques à la volée
3. Dessin de graphiques vectoriels
4. Affichage de cartes
5. Conclusion

Introduction

HTML5 offre deux nouveaux éléments graphiques :

- L'élément **<canvas>** qui permet de dessiner des graphiques, à la volée, sur une page Web par l'intermédiaire de scripts (habituellement JavaScript)
- L'élément **<svg>** qui permet de dessiner des graphiques en utilisant SVG
- Voici un exemple de dessin contenant un rectangle rouge, un rectangle dégradé, un rectangle multicolore, multicolore et un texte avec Canvas:



- En plus de ces éléments graphiques, HTML5 permet l'affichage de cartes géographiques en utilisant l'API de Google Maps.

Dessin de graphiques à la volée

Le dessin de graphique à la volée dans un document HTML se fait grâce à l'élément Canvas,

Qu'est-ce que qu'un Canvas?

- Un Canvas est une zone rectangulaire sur une page HTML, et il est spécifié avec l'élément `<canvas>`.
- L'élément `<canvas>` est juste un container pour les graphiques. Vous devez utiliser un script pour réellement tracer des graphiques.
- Il dispose de plusieurs méthodes pour les chemins de dessin, les boîtes, les cercles, des caractères et des images d'ajout.
- **Il est pris en charge par les navigateurs modernes** : IE9 +, Firefox, Opera, Chrome et Safari
- **Cependant** IE8 et antérieurs ne supportent pas l'élément `<canvas>`.

Dessin de graphiques à la volée

Comment créer un Canvas

- Voici la syntaxe pour créer un Canvas. (Par défaut, il n'a pas de bordure et pas de contenu).

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- **Remarque:** Toujours spécifier un attribut id (pour être mentionné dans un script), et les attributs width et height.
- Pour ajouter une bordure, utilisez l'attribut de style:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;"></canvas>
```

- Il est possible d'ajouter du contenu au Canvas comme
:

```
<canvas id="myCanvas" width="200" height="100">
```


Ton navigateur ne supporte pas l'élément Canvas de HTML5

```
</canvas>
```

pour les navigateurs ne prenant pas en charge l'élément Canvas

Dessin de graphiques à la volée

Comment dessiner dans un Canvas

- Tout dessin sur le Canvas doit être fait à l'intérieur d'un script JavaScript comme suit :
- Exemple

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px
solid #000000;">Ton navigateur ne supporte pas la balise Canvas de HTML5.</canvas>
<script>
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.fillStyle="#FF0000";
    ctx.fillRect(0,0,150,75);
</script>
</body>
</html>
```

Dessin de graphiques à la volée

Comment dessiner dans un Canvas

- Explication

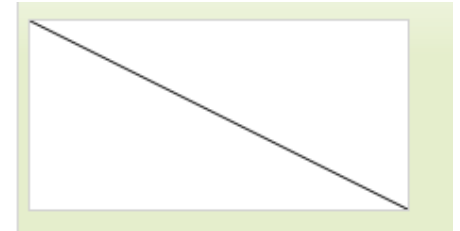
- `var c=document.getElementById("myCanvas");` permet de choisir le Canvas par son id.
- `var ctx=c.getContext("2d");` appelle sa méthode `getContext ()` (en lui passant la chaîne "2d").
- `getContext ("2d")` est un objet de HTML5 ayant de nombreuses propriétés et méthodes pour les chemins de dessin, les boîtes, les cercles, du texte, des image, etc.
- Les deux lignes suivantes dessinent un rectangle rouge:
`ctx.fillStyle="#FF0000";`
`ctx.fillRect(0,0,150,75);`
- La propriété **fillStyle** peut être une couleur CSS, un gradient, ou un motif. La valeur par défaut de `fillStyle` est `#000000` (noir).
- La méthode **fillRect** (*x,y,width,height*) dessine un rectangle rempli avec le style de remplissage.

Dessin de graphiques à la volée

Quelques exemples

- Dessiner une ligne

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.moveTo(0, 0);           //définit le point de départ de la ligne  
ctx.lineTo(200, 100);       //définit le point d'arrivée de la ligne  
ctx.stroke();                //trace les points de la ligne
```



- Dessiner un arc

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95, 50, 40, 0, 2*Math.PI); //définit un arc par les coordonnées de son centre (95,50),  
son rayon (40), son angle de départ (0) et son angle d'arrivée (2Π)  
ctx.stroke();                       //trace les points de la ligne
```



- Pour plus d'exemples, voir : https://www.w3schools.com/graphics/canvas_intro.asp

Dessin de graphiques vectoriels

Le dessin de graphiques vectoriels dans un document HTML se fait grâce à l'élément SVG,

- **Qu'est-ce que SVG?**

- SVG signifie **S**calable **V**ector **G**raphics (Graphiques vectoriels évolutifs)
- SVG est utilisé pour définir les graphiques vectoriels pour le Web
- SVG définit les graphiques au format XML
- Les graphiques SVG ne perd aucune qualité s'ils sont zoomés redimensionnés
- Les éléments et les attributs d'un élément SVG peuvent être animés
- SVG est une recommandation du W3C

- **Concurrent de SVG**

- Le principal concurrent de SVG est Flash.
- Son principal avantage sur Flash est sa conformité avec les normes comme XSL et DOM.
- Flash repose sur une technologie propriétaire qui n'est pas open source.

Dessin de graphiques vectoriels

- **Les avantages de SVG**

SVG a plusieurs avantages sur les autres formats d'image (comme JPEG et GIF) :

- Les images SVG peuvent être créées et éditées avec n'importe quel éditeur de texte
- Les images SVG peuvent être recherchées, indexées, scénarisées et comprimées
- Les images SVG s'adaptent
- Les images SVG peuvent être imprimés en haute qualité à n'importe quelle résolution
- Les images SVG sont zoomables (et l'image peut être agrandie sans dégradation)
- Internet Explorer 9 +, Firefox, Opera, Chrome et Safari supportent SVG.

- **Création d'une image SVG**

- Une image SVG peut être créée avec n'importe quel éditeur de texte, mais il est souvent plus pratique de les créer avec un programme de dessin, comme Inkscape (<https://inkscape.org>).

Dessin de graphiques vectoriels

- **Intégrer SVG dans une page HTML**

- Avec HTML5, on peut incorporer des éléments SVG directement dans votre page HTML avec l'élément `svg`,
- L'élément `<svg>` est un conteneur pour les graphiques SVG.
- SVG dispose de plusieurs méthodes pour tracer des chemins, des boîtes, des cercles, des textes et des images graphiques.

Syntaxe :

```
<!DOCTYPE html>
<html>
<body>

<svg width="100" height="100">
//Ajouter ici la méthode
</svg>

</body>
</html>
```

Dessin de graphiques vectoriels

- Quelques méthodes de SVG

- Dessiner un rectangle:

```
<svg width="400" height="100">  
  <rect x="50" y="20" width="400" height="100"  
  style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />  
</svg>
```



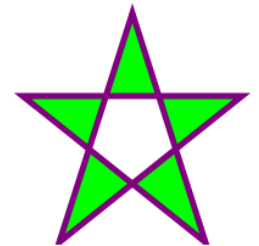
- Dessiner un cercle :

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke width="4" fill="yellow" />  
</svg>
```



- Dessiner une étoile

```
<svg width="300" height="200">  
  <polygon points="100,10 40,198 190,78 10,78 160,198"  
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />  
</svg>
```



- Pour plus de détails, voir : https://www.w3schools.com/graphics/svg_intro.asp

Comparaison des éléments graphiques

- SVG est un langage de description de graphiques 2D en XML.
- Canvas dessine des graphismes en 2D, à la volée (avec JavaScript).
- SVG est basé sur XML, ce qui signifie que chaque élément est disponible dans le DOM de SVG. Vous pouvez attacher des gestionnaires d'événements JavaScript sur un élément.
- En SVG, chaque forme dessinée est considérée comme un objet. Si les attributs d'un objet SVG sont modifiés, le navigateur peut automatiquement rendre à nouveau la forme.
- Canvas est rendu pixel par pixel. Une fois que le graphique est tiré, il est oublié par le navigateur. Si sa position doit être changée, toute la scène doit être redessiné, y compris les objets qui auraient pu être couverts par le graphique.

Comparaison des éléments graphiques

- Le tableau ci-dessous présente quelques différences importantes entre Canvas et SVG:

Canvas	SVG
<ul style="list-style-type: none">•Dépendante de la résolution•Ne supporte pas les gestionnaires d'événements•Pauvres capacités de rendu de texte•On peut enregistrer l'image résultante au format .Png ou .Jpg•Très adapté aux graphiques de jeux intensifs	<ul style="list-style-type: none">•Indépendante de la résolution•Supporte les gestionnaires d'événements•Le mieux adapté pour les applications avec des grandes zones de rendu (Google Maps)•Rendu lent si complexe (tout ce qui utilise le DOM d'un lot sera lent)•Ne convient pas aux applications de jeux

Affichage de cartes

- **Qu'est-ce que Google Maps?**



- Google Maps est un service de cartographie en ligne lancé en 2004 par Google.
- C'est un service disponible sur PC, sur tablette et sur smartphone qui permet, à partir de l'échelle d'un pays, de zoomer jusqu'à l'échelle d'une rue.
- Des prises de vue fixes montrant les détails de certaines rues sont également accessibles grâce à une passerelle vers Google Street View.

- **Concurrents de Google Maps**

- Baidu Maps, Bing Cartes, Mappy, Here offrent des services ressemblant à Google Maps.
- Mais Google Maps demeure le plus populaire avec ses services très pratiques de géolocalisation, de partage de position, de Street View, d'itinéraire, etc. qui fonctionnent sur PC et sur Smartphones.

- **L'objectif ici est de montrer comment afficher des cartes dans une page HTML en utilisant Google Maps.**

Affichage de cartes

- Les étapes à suivre

- Etape 1 : Ajout de la zone d'affiche dans le document HTML

```
<!DOCTYPE html>
<html>
<body>
    <h1>Affichage d'un carte avec Google Maps</h1>
    <div id="map">La carte sera affichée ici</div>
</body>
</html>
```

- Etape 2 : Définir la taille de la carte

```
<div id="map" style="width:400px;height:400px">
```

Affichage de cartes

- Les étapes à suivre

- **Etape 3** : Ajout de la fonction pour paramétrer les propriétés de la carte

```
<script>
  function myMap() {
    var mapOptions = {
      center: new google.maps.LatLng(51.5, -0.12),
      zoom: 10,
      mapTypeId: google.maps.MapTypeId.HYBRID
    }

    var map = new google.maps.Map(document.getElementById("map"), mapOptions);
  }
</script>
```



- **Etape 4** : Ajout de l'API Google Maps pour afficher la carte

```
<script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script>
```

- Pour plus de détails, voir : https://www.w3schools.com/graphics/google_maps_intro.asp

Conclusion

- Nous voici à la fin de la séquence 04 consacrée au graphisme dans une page web.
- Il s'agit des éléments graphiques (Canvas et SVG) introduits par HTML5 pour les besoins du graphisme et du jeu vidéo ainsi que de l'affichage de carte avec Google Maps.
- Le chapitre suivant traitera des API (Application Programming Interfaces) de HTML 5 qui sont :
 - La Géolocalisation
 - Le Drag and Drop
 - Le Stockage web local (Puissante alternative pour les cookies)
 - Les Web Workers
 - Server-sent events (SSE)