

HTML et JavaScript

Gestion des images

1) Comment manipuler des images :

Les images `<img>` sont également considérées comme des objets du DOM.

Une fois le descripteur d'objet obtenu, on peut visualiser ou changer les propriétés (taille, alignement, bordure,...).

Pour utiliser un objet image, il y a 2 méthodes :

- a) Classique : Un attribut **id** sur `<img>` et la méthode `getElementById` :

HTML :

```

```

JavaScript :

```
document.querySelector("#mon_img").width="300"; //largeur modifiée
```

- b) Directement par les objets permanents du DOM :

Les DOM tient à jour le tableau `images[]` de toutes les images de la page html.

HTML : On crée 2 images (la deuxième utilise les styles css):

```


```

JavaScript :

```
document.images[0].width="300";           // largeur de la première
document.images[1].style.width="300px";    // largeur de la seconde
document.images[0].src="autre_img.jpg" ;    // on change l'image
```

Nommer les images :

Lorsqu'il y a plusieurs images, le repérage par numéro d'index dans le tableau `images[]` peut s'avérer inconfortable. L'attribut `name` permet de donner un nom qui sera connu du DOM :

HTML :

```

```

JavaScript :

```
document.img99.width="300"; //largeur modifiée
```

2) Associer une action à une zone d'image :

En HTML, on peut ajouter à l'image un attribut `usemap`, puis définir des `<area>` qui décrivent des zones de l'image et les pages html à charger lorsqu'on clique sur ces zones :

```

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm"/>
  <area shape="circle" coords="90,58,3" href="mercur.htm"/>
</map>
```

Pour exécuter des JavaScript, il suffit de remplacer les **href** par des évènements (`onclick`, `onmouse...`).

3) pré charger les images

Quand un script affiche une image, il commence par la télécharger. Elle est ensuite conservée dans le cache du navigateur. Le premier affichage sera donc plus lent.

Solution : charger les images qu'on utilisera plus tard, dès le chargement de la page. Cela ralentit le chargement mais ensuite la navigation par JavaScript est plus fluide :

Les images sont créées et chargées dans la zone <head> :

```
<head>
  <script>
    // Prechargement des images
    img1 = new Image();
    img1.src="../images/soleil.jpg";

    img2 = new Image();
    img2.src="../images/nuage.jpg";
  </script>
</head>
```

Dans la zone <body>, on met des balises aux endroits voulus :

```
<img name="imgX" class="position1" />
<img name="imgY" class="position2" />
```

Enfin, au moment voulu, par un script, on complète des balises en indiquant l'url de l'image :

```
<script>
  document.imgX.src = img1.src;
  document.imgY.src = img2.src;
</script>
```

Note : une astuce consiste à créer les images avec une taille de 1 pixel. Le JavaScript servira à modifier leur taille au moment de l'affichage.

4) Superposer les images / le texte :

Les images font partie des objets superposable quand on utilise le paramètre CSS suivant : z-index.

Ex : `z-index:20;`

Dans cet exemple, le chiffre 20 représente un niveau au dessus du niveau de base (0). Le chiffre peut être négatif (arrière plan). Lorsqu'on superpose des objets, plus le chiffre est élevé, plus l'objet sera visible et couvrira les autres.

Sites de référence recommandés :

<http://www.w3schools.com>