

Manipuler le CSS

Éditer les propriétés CSS

Éditer les styles CSS d'un élément

La propriété style permet de modifier le css d'un élément

```
element.style; // On accède à la propriété « style » de l'élément « element »
```

Pour modifier les styles CSS il faut écrire leur nom et leur attribuer, par exemple pour width :

```
element.style.width = '150px'; // On modifie la largeur de notre élément à 150px
```

Pensez bien à écrire l'unité de votre valeur, il est fréquent de l'oublier et généralement cela pose de nombreux problèmes dans un code !

Pour accéder à une propriété CSS qui possède un nom composé il faut supprimer les tirets :

```
element.style.backgroundColor = 'blue'; // Après avoir supprimé le tiret et ajouté une majuscule au deuxième mot, le code fonctionne !
```

Récupérer les propriétés CSS

La fonction getComputedStyle()

Il n'est pas possible de récupérer les valeurs des propriétés CSS d'un élément par le biais de la propriété style vu que celle-ci n'intègre pas les propriétés CSS des feuilles de style, ce qui nous limite énormément dans nos possibilités d'analyse. Il existe une fonction permettant de remédier à ce problème : getComputedStyle().

Cette fonction va se charger de récupérer, à notre place, la valeur de n'importe quel style CSS ! Qu'il soit déclaré dans la propriété style, une feuille de style ou bien même encore calculé automatiquement, cela importe peu : getComputedStyle() la récupérera sans problème.

Toutes les valeurs obtenues par le biais de getComputedStyle() ou currentStyle sont en lecture seule. Son fonctionnement est très simple et se fait de cette manière :

```
<style>
  #text {
    color: red;
  }
</style>

<span id="text"></span>

<script>
  var text = document.getElementById('text'),
      color = getComputedStyle(text).color;
```

```
alert(color);  
</script>
```

Attention ! Les versions d'Internet Explorer antérieures à la 9 ne supportent pas la méthode `getComputedStyle()` mais la propriété `currentStyle`.

Les propriétés de type offset

Certaines valeurs de positionnement ou de taille des éléments ne pourront pas être obtenues de façon simple avec `getComputedStyle()`, pour pallier ce problème il existe les propriétés offset qui sont, dans notre cas, au nombre de cinq :

Nom de l'attribut	Contient...
<code>offsetWidth</code>	Contient la largeur complète (width + padding + border) de l'élément.
<code>offsetHeight</code>	Contient la hauteur complète (height + padding + border) de l'élément.
<code>offsetLeft</code>	Surtout utile pour les éléments en position absolue. Contient la position de l'élément par rapport au bord gauche de son élément parent.
<code>offsetTop</code>	Surtout utile pour les éléments en position absolue. Contient la position de l'élément par rapport au bord supérieur de son élément parent.
<code>offsetParent</code>	Utile que pour un élément en position absolue ou relative ! Contient l'objet de l'élément parent par rapport auquel est positionné l'élément actuel.

Leur utilisation ne se fait pas de la même manière que n'importe quel style CSS, tout d'abord parce que ce ne sont pas des styles CSS ! Ce sont juste des propriétés (en lecture seule) mises à jour dynamiquement qui concernent certains états physiques d'un élément.

Pour les utiliser, on oublie la propriété `style` vu qu'il ne s'agit pas de styles CSS et on les lit directement sur l'objet de notre élément HTML :

```
alert(el.offsetHeight); // On affiche la hauteur complète de notre élément HTML
```

Les valeurs contenues dans ces propriétés (à part `offsetParent`) sont exprimées en pixels et sont donc de type `Number`, pas comme les styles CSS qui sont de type `String` et pour lesquelles les unités sont explicitement spécifiées (px, cm, em, etc.).

La propriété offsetParent

Concernant la propriété offsetParent, elle contient l'objet de l'élément parent par rapport auquel est positionné votre élément actuel.

Lorsque vous décidez de mettre un de vos éléments HTML en positionnement absolu, celui-ci est sorti du positionnement par défaut des éléments HTML et va aller se placer tout en haut à gauche de votre page Web, par-dessus tous les autres éléments. Seulement, ce principe n'est applicable que lorsque votre élément n'est pas déjà lui-même placé *dans* un élément en positionnement absolu. Si cela arrive, alors votre élément se positionnera non plus par rapport au coin supérieur gauche de la page Web, mais par rapport au coin supérieur gauche *du précédent élément placé en positionnement absolu, relatif ou fixe*.

Les propriétés offsetTop et offsetLeft contiennent le positionnement de votre élément *par rapport à son précédent élément parent* et non pas par rapport à la page ! Si nous voulons obtenir son positionnement par rapport à la page, il faudra alors aussi ajouter les valeurs de positionnement de son (ses) élément(s) parent(s).

Voici le problème mis en pratique ainsi que sa solution :

```
<style>
  #parent,
  #child {
    position: absolute;
    top: 50px;
    left: 100px;
  }

  #parent {
    width: 200px;
    height: 200px;
    background-color: blue;
  }

  #child {
    width: 50px;
    height: 50px;
    background-color: red;
  }
</style>

<div id="parent">
  <div id="child"></div>
</div>

<script>
  var parent = document.getElementById('parent');
  var child = document.getElementById('child');

  alert("Sans la fonction de calcul, la position de l'élément enfant est : \n\n" +
    'offsetTop : ' + child.offsetTop + 'px\n' +
    'offsetLeft : ' + child.offsetLeft + 'px');

  function getOffset(element) { // Notre fonction qui calcule le positionnement complet
```

```

    var top = 0,
        left = 0;

    do {
        top += element.offsetTop;
        left += element.offsetLeft;
    } while (element = element.offsetParent); // Tant que « element » reçoit un «
offsetParent » valide alors on additionne les valeurs des offsets

    return { // On retourne un objet, cela nous permet de retourner les deux valeurs
calculées
        top: top,
        left: left
    };
}

alert("Avec la fonction de calcul, la position de l'élément enfant est : \n\n" +
    'offsetTop : ' + getOffset(child).top + 'px\n' +
    'offsetLeft : ' + getOffset(child).left + 'px');
</script>

```

En HTML, la largeur (ou hauteur) *complète* d'un élément correspond à la valeur de width + celle du padding + celle des bordures.

Par exemple, sur ce code :

```

<style>
    #offsetTest {
        width: 100px;
        height: 100px;
        padding: 10px;
        border: 2px solid black;
    }
</style>

<div id="offsetTest"></div>

```

La largeur complète de notre élément <div> vaut : 100 (width) + 10 (padding-left) + 10 (padding-right) + 2 (border-left) + 2 (border-right) = **124px**.

Et il s'agit bien de la valeur retournée par offsetWidth :

```

var offsetTest = document.getElementById('offsetTest');
alert(offsetTest.offsetWidth);

```