

Techniques pour le développement d'interfaces

1. création de graphiques SVG

Introduction à XML

Définition

XML, pour eXtensible Markup Language (langage de balisage extensible), est un langage de balisage généraliste recommandé par le W3C comme l'est HTML. XML est un sous-ensemble du langage SGML. Cela signifie que contrairement aux autres langages de balisages, XML n'est pas prédéfini, vous devez définir vos propres balises. Le but principal de ce langage est le partage de données entre différents systèmes, tel qu'Internet.

De nombreux autres langages sont basés sur XML, comme par exemple XHTML, MathML, SVG, XUL, XBL, RSS et RDF. Vous pouvez créer votre propre langage basé sur XML.

Du XML « correct » (valide et « bien formé »)

Pour être correct, un document XML doit être « bien formé », se conformer à toutes les règles de syntaxe du XML, et « valide », se conformer aux règles d'un langage spécifique.

La plupart des navigateurs, dont Mozilla, offrent un outil de validation qui informera souvent lors de la lecture ou de l'affichage de documents mal formés.

Exemple

Voici un exemple de document mal formé : un des éléments possédant une balise d'ouverture, <warning>, n'a pas de balise de fermeture et ce n'est pas un élément vide (ou auto-fermant comme les balises
 ou en XHTML).

```
<code>
<message>
  <warning>
    Hello World
  </message>
</code>
```

<https://www.w3shcools.com/graphics/svginhtml.asp>

2. Usage de l'API vibration

API de Vibration

La vibration est décrite comme une série de marche-arrêt des impulsions, qui peuvent être de durées différentes. La série peut être soit un entier décrivant le nombre de millisecondes à vibrer ou un tableau d'entiers décrivant une série de pauses et de vibrations. La vibration est contrôlée par une seule méthode

Vous pouvez effectuer une seule vibration du matériel en spécifiant soit une valeur de type "entier", soit un tableau d'entiers.

3. Les gestionnaires d'événements tactiles

Événements tactiles / Touch events

Afin de fournir un support de qualité pour les interfaces tactiles, les événements tactiles (touch events) permettent d'interpréter les interactions tactiles (sur les écrans ou trackpads).

[Définitions](#)

Surface

La surface tactile. Cela peut être un écran ou un trackpad.

Point de toucher (Touch point)

Le point de contact avec la surface. Cela peut être un doigt ou un stylet (ou un coude, une oreille, un nez... enfin il y a quand même des chances que cela soit un doigt).

[Interfaces](#)

[TouchEvent \(en-US\)](#)

Représente l'événement qui se produit quand l'action de toucher change.

[Touch \(en-US\)](#)

Représente un point unique de contact entre l'utilisateur et la surface tactile.

[TouchList \(en-US\)](#)

Représente un groupe de plusieurs interactions tactiles. Cela peut par exemple être le cas quand l'utilisateur utilise plusieurs doigts pour toucher simultanément la même surface.

[DocumentTouch](#)

Contient des méthodes permettant de créer les objets [Touch \(en-US\)](#) et [TouchList \(en-US\)](#).

[Exemple](#)

Cet exemple permet de gérer un toucher multiple (plusieurs contacts simultanés), permettant ainsi à l'utilisateur de dessiner dans un [<canvas>](#) avec plusieurs doigts. Cela ne fonctionne qu'avec les navigateurs supportant les interactions tactiles.

Note : Le texte qui suit utilisera le terme de « doigt » pour désigner le point de toucher entre l'utilisateur et la surface. Bien entendu, cela est transposable avec une autre méthode de toucher (stylet...).

[Initialiser les gestionnaires d'événements](#)

Quand la page charge, la fonction startup() décrite ci-dessous est appelée par l'attribut onload de l'élément [<body>](#).

```
function startup() {  
  var el = document.getElementsByTagName("canvas")[0];  
  el.addEventListener("touchstart", handleStart, false);  
  el.addEventListener("touchend", handleEnd, false);  
  el.addEventListener("touchcancel", handleCancel, false);  
  el.addEventListener("touchleave", handleLeave, false);  
  el.addEventListener("touchmove", handleMove, false);  
}
```