

Exercice6 : DOJO dialogue

Créer un projet Exercice6. Quand les bibliothèques comme jQuery et Prototype axent sur l'accès au modèle objet de document (DOM), Dojo crée une couche riche sur le dessus de la DOM.

Etape1 : préparation

Voici le squelette de code.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <script type="text/javascript"
src="http://o.aolcdn.com/dojo/1.2.0/dojo/dojo.xd.js" djConfig="parseOnLoad:
true"></script>

  <script type="text/javascript">

    // JavaScript and Dojo calls will go here

  </script>

  <style type="text/css">

    @import "http://o.aolcdn.com/dojo/1.2.0/dijit/themes/tundra/tundra.css";

  </style>
</head>
<body class="tundra">

<!-- HTML and Dojo tags will go here-->

</body>
</html>
```

Comme avec importe quel framework JavaScript ou bibliothèque, la première chose que nous avons à faire est d'inclure le fichier de code.

Avec Dojo, vous pouvez charger un fichier unique de AOL Content Delivery Network (CDN), et ne pas héberger le fichier vous-même. Le fichier sera mis en cache de façon appropriée et vous n'avez pas à gérer le téléchargement du code.

Etape3 : création de bouton

Il est demandé de créer un bouton. Le thème Tundra CSS prend soin de ce qu'il ressemble. Dijits, la partie widget du Toolkit Dojo, est utilisé pour transformer le bouton dans un bouton spécial.

La première chose que vous devez faire est de vous assurer que le code qui produit le bouton est chargé. Le lien vers le fichier JavaScript Dojo que nous avons inclus plus tôt, prend soin de cela pour vous, mais depuis Dojo ne vous donne que ce que vous avez besoin, vous devez dire ce que vous voulez.

```
dojo.require("dijit.form.Button");
```

Maintenant Dojo est prêt à créer le bouton, mais nous devons montrer où dans notre HTML. Placez ce balisage où vous souhaitez le bouton pour afficher:

```
<button dojoType="dijit.form.Button" id="buttonTest">
  Clicky
  <script type="dojo/method" event="onClick">
    alert('Clicky-click! You clicked and now you see this box');
  </script>
</button>
```

Sauvegarder et recharger votre fichier dans un navigateur. Vous devriez voir un bouton normal pour un moment, puis un bouton carré avec une légère pente.

Etape4 : création de bouton avec une meilleure boîte de dialogue

Dojo a une zone meilleure d'alerte, appelé Dialog. Dans cette section, nous allons déclencher cette nouvelle boîte d'alerte avec le même bouton que nous avons utilisé sur la section précédente.

```
dojo.require("dijit.Dialog");
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <script type="text/javascript"
src="http://o.aolcdn.com/dojo/1.2.0/dojo/dojo.xd.js" djConfig="parseOnLoad:
true"></script>
  ...
</head>
<body class="tundra">
  ...
</body>
</html>
```

Maintenant, vous pouvez supprimer cette ligne d'alerte de la balise script dojo / méthode. Où vous avez l'habitude d'avoir la ligne d'alerte, la remplacer par cette ligne:

```
dijit.byId('dialogTest').show();
```

C'est l'objet qui deviendra notre dialogue. Collez ce balise <div> spéciale ci-dessous sur le bouton (mais avant la balise body fin).

```
<div dojoType="dijit.Dialog" id="dialogTest" title="Clicky-click!">Woohoo! You clicked and now you see
this box.</div>
```

Le dojoType peut sembler familier du code du bouton, qui a utilisé une syntaxe similaire. Dans ce cas, le type est dijit.Dialog. Nous avons également donné la div un id, dialogTest, qui est le même que ce que nous sommes passés à dijit.byId ci-dessus.

Etape5 : faire une boîte de dialogue par programmation

À ce stade, vous avez vu comment Dojo peut changer la façon dont les éléments d'interface ont l'air. Le bouton Dojo et Dialog étaient tous deux créés complètement avec le code HTML comme, en s'appuyant sur Dojo. Pour avoir une fonction vraiment utile de boîte d'alerte, nous ne voulons pas le jeter dans un div spéciale chaque fois que nous voulons afficher un message. Pour cela, nous avons besoin d'une boîte d'alerte programmable!

```
function dialogAlert(txtTitle, txtContent)
{
...
}
```

Nous pouvons appeler cette nouvelle fonction aussi souvent que nous aimons, autant que nous appelions la fonction d'alerte. Au lieu d'un seul message, ce message d'alerte utilise également un titre. Maintenant, nous devons appeler cette nouvelle fonction. Remplacez le code dans l'événement onclick avec cette ligne:

`dialogAlert('Clicky-click!', 'Woohoo! You clicked and now you see this box.');`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <script type="text/javascript"
src="http://o.aolcdn.com/dojo/1.2.0/dojo/dojo.xd.js" djConfig="parseOnLoad:
true"></script>
  <script type="text/javascript">
    dojo.require("dijit.form.Button");
    dojo.require("dijit.Dialog");

    function dialogAlert(txtTitle, txtContent)
    {
...
    }
  </script>
  <style type="text/css">
    @import "http://o.aolcdn.com/dojo/1.2.0/dijit/themes/tundra/tundra.css";
  </style>
</head>
<body class="tundra">
...
</body>
</html>
```