

Slider JavaScript

Voici un code tout simple pour fabriquer un slider en JavaScript. Le principe est de changer la source des images en cliquant sur les flèches latérales.

Code HTML : slider.html

Au premier chargement de la page, le code HTML fait apparaître la première image du slider. Le slider est contenu dans une balise `<div id= »slider »>` Qui contient l'image du premier slide suivi des flèches gauche et droite qui sont elles aussi des balises `<div>`.

```
<!doctype html>
<html lang="fr">

<head>
  <meta charset="utf-8">
  <title>Slider</title>
  <link rel="stylesheet" href="slider.css">
  <script src="slider.js"></script>

</head>

<body>
  <h1>Le beau slider</h1>
  <div id="slider">
    
    <div id="precedent" onclick="ChangeSlide(-1)"><<</div>
    <div id="suivant" onclick="ChangeSlide(1)">>></div>
  </div>
</body>

</html>
```

Le code HTML est relié à une feuille de style externe *slider.html* et un code JavaScript externe *slider.js*.

Code CSS : slider.css

Le code CSS permet de centrer horizontalement le slider et de faire apparaître progressivement les flèches, lors du survol de la souris, grâce à la propriété *transition* qui va effectuer les changements d'opacité sur une durée de 300 millisecondes. Par conséquent, le survol du slider, fera apparaître les flèches en passant de l'*opacity*: 0 à 1 en 0.3 secondes.

```
body {
  font-family: "arial", sans-serif;
  text-align: center;
}

#slider {
  position: relative;
  width: 1200px;
  margin: auto;
  display: flex;
  align-items: center;
}

#precedent,
#suivant {
  cursor: pointer;
  transition: opacity 0.3s ease;
  opacity: 0;
}
```

```

    position: absolute;
    font-size: 100px;
    color: rgba(220, 220, 220, 0.8);
    background-color: rgba(0, 0, 0, 0.8);
    padding: 10px;
}

#precedent {
    left: 0;
}

#suivant {
    right: 0;
}

#slider:hover #precedent,
#slider:hover #suivant {
    opacity: 1;
}

```

Le conteneur *#slider* est en position relative, ce qui permet à ses enfants de se positionner en *absolute* par rapport à lui. Il ne faut pas oublier cette règle CSS : un positionnement *absolute* se fait toujours dans le cadre du premier bloc parent positionné. Les flèches seront donc plaquées contre les bords grâce au positionnement *absolute*.

Le centrage vertical des flèches est géré par les propriétés *display: flex* et *align-items: center*.

Code JavaScript : slider.js

Grâce aux évènements *onclick* sur les flèches latérales, le code JavaScript lance la fonction *ChangeSlide()* en passant le paramètre *-1* pour la flèche gauche et *1* pour la flèche droite.

Pour ajouter un slide, il suffit de compléter la variable de type tableau *slide* en ajoutant un nom d'image correspondant au slide.

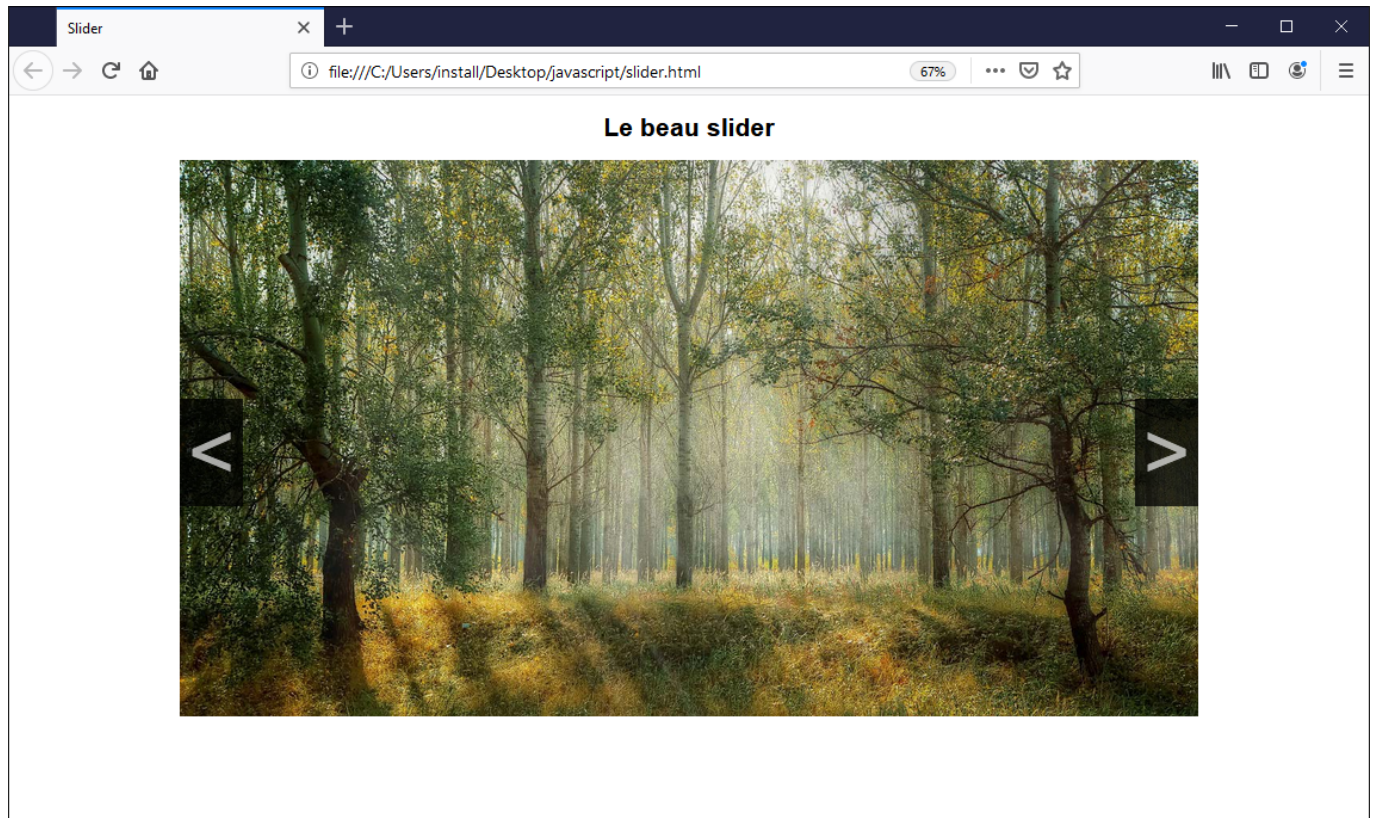
```

const slide = ["foret-peuplier.jpg", "paysage-montagne.jpg", "chemin-automne.jpg", "prairie-
alpes.jpg"];
let numero = 0;

function ChangeSlide(sens) {
    numero = numero + sens;
    if (numero < 0)
        numero = slide.length - 1;
    if (numero > slide.length - 1)
        numero = 0;
    document.getElementById("slide").src = slide[numero];
}

```

Lorsque l'utilisateur clique sur la flèche suivante, le paramètre *sens* est égal à *1* et la variable *numero* est incrémentée. L'instruction conditionnelle *if* permet de vérifier qu'on ne dépasse pas le nombre d'images du slide. Et si on dépasse, on réinitialise la variable *numero*, ce qui permet de boucler le slider. *slide.lenght* permet de compter le nombre d'images du tableau *slide*. on retire *-1* car la première image du slide est *slide[0]*.



Encore un slider magnifique inventé par LesDocs ☐

Défilement automatique

Pour ajouter la fonctionnalité diaporama à notre slider, on peut ajouter la ligne JavaScript suivante :

```
setInterval("ChangeSlide(1)", 4000);
```

La méthode *setInterval()* va appeler la fonction *ChangeSlide* avec le paramètre 1 toutes les 4 secondes soit 4000 millisecondes. C'est un peu comme si quelqu'un cliquait sur la flèche de droite toutes les 4 secondes.

La vidéo

La fabrication de ce slider est expliquée en détail dans cette vidéo. Bon visionnage !