

R213 - TD/TP - Séances 2 et 3 - Le slider

<http://tinyurl.com/MMIChamps-R213-03>

L'objectif de ce TP est de créer un « slider ».

Deux techniques sont proposées :

- la première, **faire défiler une bande d'images**, est plus simple en termes de Javascript, elle ne fait que des manipulations du DOM et permet de défiler automatiquement vers la gauche, ou manuellement vers la droite, une bande d'images dont seule une partie (correspondant à la largeur d'une image) est montrée ;
- la deuxième, **charger et déplacer deux images**, propose de stocker en permanence le numéro de l'image que l'on est en train d'afficher et celle qui suit ou qui précède : à chaque fois que l'on souhaite déclencher un mouvement d'image vers la gauche, on charge l'image suivante, on déplace progressivement les deux images pour montrer l'image suivante, puis on réinitialise le bloc des deux images pour préparer le déplacement de l'image suivante.

Si vous avez besoin d'images types pour votre « slider », vous pouvez utiliser le site Lorem Picsum à l'adresse <https://picsum.photos/>

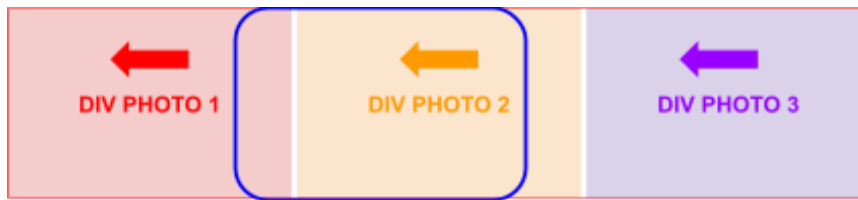
Première solution, faire défiler une bande d'images

Le principe de ce slider est qu'il s'agit d'un cadre (la « fenêtre » en bleu ci-dessous, ayant la classe `js-slider`) qui contient une bande d'images alignées les unes à côté des autres. Si cette bande d'images (ayant la classe `js-photos`) contient trois photos, elle est de largeur égale à trois fois la largeur de la fenêtre. La largeur de chaque photo de la bande d'images est égale à la largeur de la fenêtre. Toutes les hauteurs des photos sont égales. Pour faciliter les choses (si vous voulez par exemple ajouter un texte sur chaque photo), il peut être utile de mettre chaque photo dans son propre bloc `div` (ayant chacun la classe `js-photo`).

Étape 1) Avant le lancement de la première animation, la propriété CSS `left` du bloc `div` contenant les trois photos vaut 0 :



Étape 2) Pendant l'animation, le bloc div contenant les trois photos est translaté vers la gauche, en faisant varier sa propriété CSS `left` :



Étape 3) Après l'animation :



Q1. Préparez le code HTML et le code CSS d'un bloc de classe `js-slider`, qui contient un bloc div de classe `js-photos`, qui contient tous les blocs div de classe `js-photo`, côte à côte, que vous souhaitez faire défiler, organisés comme montré dans le schéma de l'étape 1.

Assurez vous que tout ce qui dépasse du bloc de la classe `js-slider` est caché.

Indication à surligner si vous ne savez pas faire la fin de la phrase en CSS :

Q2. Écrivez une fonction javascript `decaleGauche` qui sélectionne le bloc de classe `js-photos` et qui décale son bord gauche d'une largeur égale à la largeur du bloc de classe `js-slider`. Appelez cette fonction pour voir si elle s'exécute bien.

Q2.bis Faites en sorte que ce décalage vers la gauche devienne dynamique à l'aide d'une variable `position` que vous initialisez à 0 et que vous incrémentez dans `decale gauche`.

Q3. Ajoutez en Javascript un minuteur permettant de déclencher automatiquement un appel de cette fonction toutes les 3 secondes : le déplacement de l'image actuelle vers la gauche et son remplacement par l'image suivante. Pensez à bien gérer le cas particulier de la dernière image.

Indication à surligner si vous ne connaissez pas les minuteurs en Javascript :

Q4. Ajoutez dans votre fonction `decaleGauche` un test qui vérifie si le bloc div de classe `js-photos` est situé le plus à gauche possible, c'est-à-dire que c'est la photo la plus à droite qui est affichée : dans ce cas, soit vous arrêtez le décalage, soit vous faites glisser le bandeau vers la droite pour revenir à la première photo. Pensez à créer une fonction `retourDebut` pour segmenter le code.

Q5. Ajoutez dans votre code HTML un bouton qui permettra de forcer manuellement le défilement d'une photo vers la gauche.

Q6. Ajoutez un code Javascript permettant de détecter le clic sur ce bouton et de lancer un appel de la fonction `decaleGauche`.

Q7. Ajoutez un bouton et le code Javascript associé pour permettre aussi de faire manuellement des décalages vers la droite. Vous créerez donc le code HTML de ce bouton, une nouvelle fonction `decaleDroite` inspirée de `decaleGauche` (attention à ce qui se passe quand vous arrivez à la première image, c'est-à-dire que le bloc `div` de classe `js-photos` est situé le plus à droite possible), et appellerez `decaleDroite` au clic sur ce bouton.

Aller plus loin avec la première solution : créer un slider infini avec des clones

Q8. Créez deux nouvelles slides de part et d'autre du slider pour simuler un slider infini. Créez une nouvelle `div js-photo` que vous ajoutez **avant** toutes les autres et insérez les éléments de la **dernière slide** dedans. Créez ensuite une nouvelle `div js-photo` que vous ajoutez **après** toutes les autres et insérez les éléments de la **première slide** dedans.

Pour mieux comprendre ce qu'il se passe, je vous conseille d'enlever temporairement le `overflow:hidden` de `js-slider`. Vous le remettrez en place à la fin.

Q9. Changez dans votre code javascript la position initiale du slider à 1 à la place de 0. Et changez également dans le code CSS la première valeur `left` de `js-photos` à `-500px`. Ceci vous permettra d'être positionné sur la première slide réelle, et non la slide clonée.

Q10. Ajustez la fonction `retourDebut` pour, dans un premier temps, ne plus avoir de transition lors du passage sur la slide clonée du début. Pour enlever l'animation de sliding, appliquez la valeur `"0s"` à la propriété CSS `transition` du bloc `js-photos` au tout début de la fonction `retourDebut` avant d'effectuer le mouvement en left du slider vers le clone.

Q.11 Après le saut sur le clone, slidez vers la slide réelle 1. Pour ce faire, directement après ce saut, ajoutez un appel de fonction `setTimeout` de *20 millisecondes* et placez-y votre ancien code qui slidait vers la slide réelle (dans ce `setTimeout`, changez également une nouvelle fois la valeur de la propriété `transition` pour voir à nouveau les animations de transition).

Q.12 Appliquez le même procédé pour la fonction `retourFin`.

Aller plus loin avec la première solution : ajouter des « dots »

Les « dots » permettent d'obtenir un lien direct vers les photos du slider de la façon suivante : un ensemble de petits points, les « dots », sont affichés de façon horizontale au bas du slider; un clic sur le n -ième point permet de faire défiler les photos jusqu'à arriver à la n -ième photo.

Un schéma d'illustration de ce concept est fourni [à cette adresse](#).

- Commencez par créer votre liste de dots en HTML et désignez-les en CSS.
- Récupérer en Javascript un tableau de dots à l'aide de la fonction `querySelectorAll`
- Parcourez ce tableau à l'aide d'un :

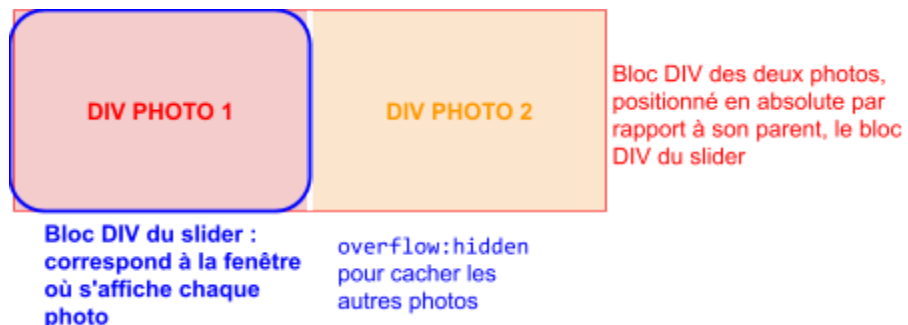
```
forEach(function(element, index) {  
  
})
```

- Remarquez que la boucle `forEach` vous donne accès à l'élément en cours ET à son `index` dans le tableau
- Ajouter un écouteur d'événement `click` à l'élément en cours de parcours afin que lorsque l'événement `click` soit détecté vous puissiez vous aider de la variable `index` pour changer votre variable `position` et positionner de la bonne manière le bloc `div` de classe `js-photos`

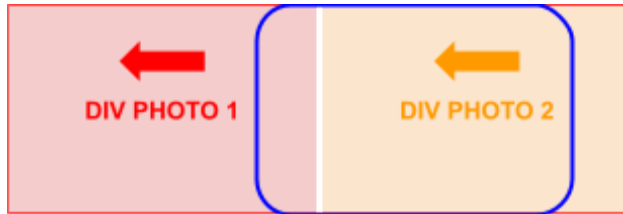
Deuxième solution, charger et déplacer deux images

Le principe de ce slider est qu'il s'agit d'un cadre (la « fenêtre » en bleu ci-dessous) qui contient un bloc de largeur égale à deux fois la largeur de la fenêtre, contenant deux blocs d'images alignés l'un à côté de l'autre, l'une de la photo de gauche, l'autre de la photo de droite.

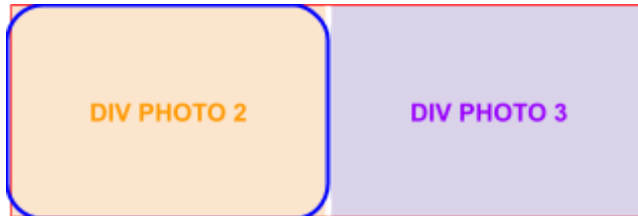
Étape 1) Avant le lancement de la première animation, la propriété CSS `left` du bloc `div` contenant les deux photos vaut 0 :



Étape 2) Pendant l'animation, le bloc `div` contenant les deux photos est translaté vers la gauche, en faisant varier sa propriété CSS `left` :



Étape 3) Après l'animation, on modifie la div de la photo de gauche pour y insérer la photo actuellement visible (la photo 2, donc) et la div de la photo de droite pour insérer la photo suivante (la 3, donc) et on redécalle instantanément vers la droite le bloc des deux photos.



Q1. Préparez le code HTML et le code CSS d'un bloc de classe `js-slider`, qui contient un bloc `div` de classe `js-photos`, qui contient un bloc `div` de classe `js-gauche` et un bloc `div` de classe `js-droite`, côte à côte, organisés comme montré dans le schéma de l'étape 1 et préparez des blocs cachés ayant tous la classe `js-photo` qui contiennent les photos que vous souhaitez faire défiler.

Assurez vous que tout ce qui dépasse du bloc de la classe `js-slider` est caché.

Indication à surligner si vous ne savez pas faire la fin de la phrase en CSS :

Q2. Écrivez une fonction javascript `decaleGauche` qui sélectionne le bloc de classe `js-photos` et qui, à l'aide d'une animation, décale progressivement son bord gauche d'une largeur égale à la largeur du bloc de classe `js-slider`.

Q3. Créez une variable `n` qui contient le numéro du bloc `div` de la photo de classe `js-photo` qui se trouve dans le bloc `div` de classe `js-gauche` (par exemple 0 s'il s'agit de la première photo) puis mettez à jour le bloc `div` de classe `js-photos` pour préparer la prochaine animation : mettez dans le bloc `div` de classe `js-gauche` le contenu du bloc `div` de classe `js-droite` puis mettez à jour `n` puis insérez dans le bloc `div` de classe `js-droite` le contenu du `n`-ième bloc `div` de classe `js-photo`.

Attention, si on est arrivé à la dernière photo à cette étape (c'est-à-dire que la `n`-ième image n'existe pas, et donc le sélecteur renvoie un résultat vide), il faut mettre à jour `n` en le réinitialisant à la valeur 0 (la prochaine image qui sera affichée sera la première image, puisqu'on est arrivé à la dernière)

Q4. Ajoutez en Javascript un minuteur permettant de déclencher automatiquement un appel de cette fonction toutes les 3 secondes : le déplacement de l'image actuelle vers la gauche et son remplacement par l'image suivante.

Indication à surligner si vous ne connaissez pas les minuteurs en Javascript :

Q5. Ajoutez dans votre code HTML un bouton qui permettra de forcer manuellement le défilement d'une photo vers la gauche.

Q6. Ajoutez un code Javascript permettant de détecter le clic sur ce bouton et de lancer un appel de la fonction `decaleGauche`.

Q7. Complétez vos codes HTML et Javascript pour permettre aussi de faire manuellement des décalages vers la droite. Votre bloc div de classe `js-photos` contiendra maintenant 3 photos, auxquelles je vous conseille de donner les noms de classe `js-gauche`, `js-centre` et `js-droite`.