



Master Bioinformatique
Programmation événementielle 2017/2018
Travaux Pratiques

TP
Javascript

NB : Commencez par récupérer l'archive du TP2. Cette archive contient des images utilisées dans la suite du TP.

Exercice 1 : Gestion d'événements en Javascript

- L'objectif de cet exercice est de prendre en main Javascript au travers de la gestion de deux événements :
- Changement d'image lorsque l'utilisateur déplace sa souris sur l'image
 - Modification du texte lors d'un clic sur le texte.

Exercice 1



Premier texte sans italique.

FIGURE 1 – Exercice 1

HTML/CSS

Question 1 : Créer une page Web (HTML+CSS) qui permette d'obtenir un rendu proche de celui présenté en Figure 1. Au chargement de la page, l'image qui s'affiche est l'image 1 . jpg et le texte ne contient pas d'italique.

Javascript

Question 2 : Créer une fonction `setupListeners`. Appeler cette fonction au chargement de la page Web. Cette fonction contiendra les abonnements des fonctions aux éléments (`addEventListener`).

Question 3 : Lors d'un clic sur le texte :

1. remplacer son contenu par "Nouveau texte en italique"
2. formatez le nouveau contenu de façon à ce qu'il apparaisse en italique

Question 4 : Lorsque le pointeur de la souris se déplace sur l'image, elle doit être remplacée par l'image 2. jpg. L'événement à capturer est de type `mouseover`.

Question 5 : Lorsque le pointeur de la souris est en dehors de l'image, l'image initiale doit être restaurée (1. jpg). L'événement à capturer est de type `mouseout`.

Exercice 2 : Carrousel d'images

Le but de cet exercice est de créer un carrousel, c'est-à-dire un défilement d'images selon deux événements :

1. le clic sur un bouton précédent/suivant
2. après que 2 secondes se sont écoulées.

Carousel

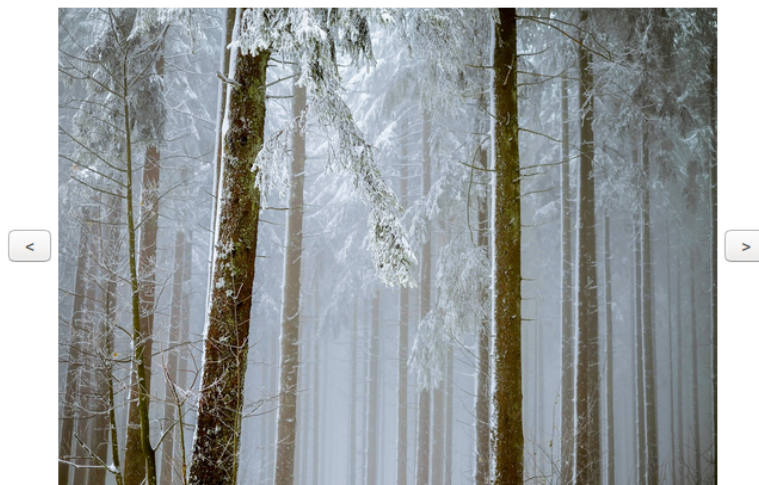


FIGURE 2 – Carrousel

HTML/CSS

Question 1 : Créer une page Web (HTML+CSS) qui contient deux boutons "précédent" et "suivant". Spécifier deux identifiants distincts pour ces boutons.

Question 2 : Placer l'image entre les deux boutons. L'image qui s'affiche au chargement de la page est 1. jpg du répertoire `imgCarrousel`. Donner un identifiant à l'image.

Question 3 : Créer un fichier CSS de façon à ce que le rendu soit proche de celui de la Figure 2.

Javascript

Question 4 : Définir la fonction `setupListeners`.

Question 5 : Stocker les chemins des images 1. jpg, 2. jpg et 3. jpg dans un tableau (Array).

Question 6 : Effectuer le défilement de l'image au clic sur les boutons "précédents" et "suivants".

Question 7 : Effectuer le défilement automatique de l'image après 2 secondes d'inactivité, grâce à la fonction `setTimeout`. Documentez-vous au besoin sur l'utilisation de cette fonction.

Exercice 3 : Dessin dans une page HTML

Le but de cet exercice est de dessiner des formes simples (lignes, rectangles, cercle...) dans une page Web (cf. Figure 3).

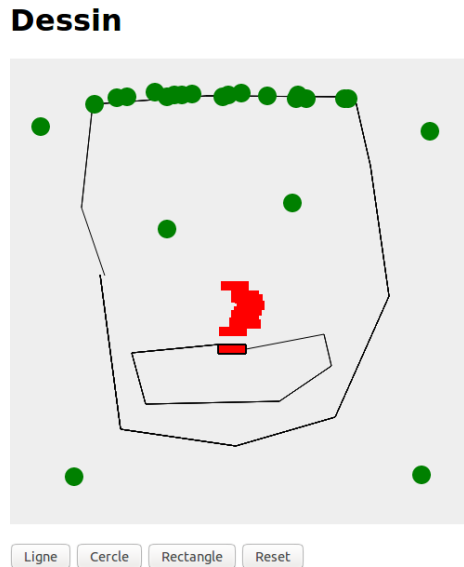


FIGURE 3

HTML/CSS

Question 1 : Créer une page Web (HTML+CSS) qui contient une zone de dessin : il s'agit de la balise `<canvas>`. Préciser la largeur et la hauteur :

```
<canvas id="myCanvas" width="500" height="500" style="background-color:#EEEEEE;"></canvas>
```

Question 2 : Placer différents boutons permettant de choisir le mode de dessin : cercle, rectangle, ligne.

Javascript

Préambule Le canvas correspond à la zone de dessin dans la page Web. Du côté Javascript, il est nécessaire de récupérer le **contexte**, c'est-à-dire une interface avec des fonctions dédiées au dessin.

Pour récupérer le contexte, il suffit de faire :

```
var contexte = canvas.getContext("2d");
```

Différentes méthodes sont associées au contexte. Chaque nouveau dessin, ou "chemin", doit débuter par la méthode `beginPath()` et être terminé par `closePath()`. Les méthodes telles que `moveTo()`, `lineTo()`, `arcTo()` sont utilisées pour ajouter des segments au chemin. Après l'appel à ces méthodes, le chemin est créé, mais ne s'affiche pas sur le canvas : il est nécessaire d'appeler les méthodes `fill()` ou `stroke()` pour le dessiner.

Les méthodes sont définies pour l'objet `contexte` et peuvent donc être appelées de la façon suivante :

```
contexte.nomMethode();
```

Utilisez la documentation disponible sur Internet pour le détail des paramètres passés à ces méthodes.

Question 3 : Implémenter la fonction de dessin de segments :

1. Le premier clic définit la position (x,y) du début du segment.
2. Au second clic, une ligne doit être tracée entre le point (x,y) et le second point cliqué.

3. De façon générale : au n -ième clic, une ligne doit être tracée entre le point $n - 1$ et le point n .

Pour cela :

1. Récupérer la position du clic
2. **S'il ne s'agit pas du premier point**, dessiner une ligne par les méthodes `lineTo()` et `moveTo()`.

Question 4 : Implémenter la fonction de dessin de cercle, en utilisant la méthode `arc()`. Le cercle doit être centré au point cliqué. Le dessin du cercle ne doit être activé que lorsque l'utilisateur clique sur le bouton "cercle", le mode par défaut de dessin étant le segment.

Question 5 : Implémenter la fonction de dessin de rectangle, en utilisant la méthode `rect()`. Le rectangle doit être centré au point cliqué. Le dessin du rectangle ne doit être activé que lorsque l'utilisateur clique sur le bouton "rectangle".

Question 6 : Implémenter une fonction qui permet de remettre à zero la zone de dessin, en utilisant la méthode `clearRect()`.