

TD3: gestion d'événements

Exercice 1 :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Gestion d'événements</title>
</head>
<body>
  <h1 id="titre">Gestion d'événements</h1>
  <p id="p1" class="paragraphe" >
    
    <span>Légende de l'image 1</span>
  </p>
  <p id="p2" class="paragraphe" >
    
    <span>Légende de l'image 2</span>
  </p>
  <script>
  </script>
</body>
</html>
```

Ecrire les instructions suivante :

- Ecrire l'instruction qui enregistre dans une constante l'élément html ayant l'id p1.
- Ecrire l'instruction qui enregistre dans une constante les éléments html ayant la classe paragraphe.
- Ecrire l'instruction qui enregistre dans une constante les éléments html .
- Ecrire l'instruction qui enregistre dans une constante les éléments html descendants directs de l'élément ayant l'id p1.
- Ecrire l'instruction qui ajoute la classe « light » à l'élément html <body>.

Indiquer pour chaque extrait de code, la cible, l'événement et l'action.

A)

```
const target = document.querySelector('h1');
target.addEventListener('mouseover', action);
function action(event) {
  console.log( "survol d'un titre" );
}
```

| | |
|-------------------|--|
| l'événement | |
| la cible/target | |
| l'action associée | |

B)

```
const target = document.getElementById('p1');
target.addEventListener('click', action);
function action(event) {
    console.log( event.target );
}
```

| | |
|-------------------|--|
| l'événement | |
| la cible/target | |
| l'action associée | |

C)

```
const target = document.getElementById('img1');
target.addEventListener('click', action);
function action(event) {
    console.log( event.target.src );
}
```

| | |
|-------------------|--|
| l'événement | |
| la cible/target | |
| l'action associée | |

Corriger les fautes dans les instructions suivantes

```
const cible = document.querySelector('h1#titre');
cible.addEventListner('mouseover', 'action');
function action( ) {
    console.log( "event.target" );
}
```

Exercice 2 : Compléter les phrases suivantes en vous aidant de la documentation

<https://developer.mozilla.org/fr/docs/Web/Events>

a) Souris

1. L'évènement est émis lorsqu'un bouton d'une souris est pressé puis relaché.
2. L'évènement est émis lorsqu'un bouton d'une souris est pressé sur la cible.
3. L'évènement est émis lorsqu'un bouton d'une souris est relaché sur la cible.
4. L'évènement est émis lorsque la souris est déplacée au-dessus de la cible.
5. L'évènement est émis lorsque la souris est déplacée en dehors de la cible.
6. L'évènement est émis lorsque la souris rentre au-dessus de la cible.
7. L'évènement est émis lorsque l'on fait défiler le document.

b) Formulaire

1. L'évènement est émis lorsque l'on modifie une balise `<select></select>`
2. L'évènement est émis lorsque l'on donne le focus à une balise `<input>`
3. L'évènement est émis lorsque l'on clique sur le bouton « envoyer » d'un formulaire

c) Clavier

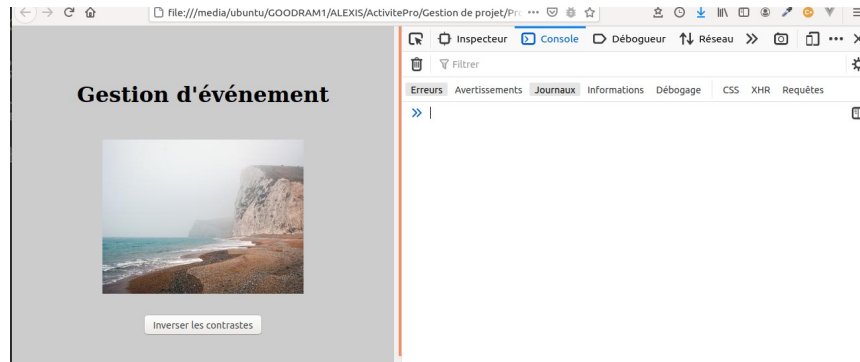
1. L'évènement est émis lorsque l'on presse une touche du clavier
2. L'évènement est émis lorsque l'on relâche une touche du clavier

d) Chargement de la page

L'évènement est émis lorsque la page HTML est complètement chargée ainsi que toutes les ressources (le CSS et les images). La cible de l'évènement est

Exercice 3:

Récupérer le fichier ex3.html ; l'ouvrir dans votre éditeur de code ainsi que dans votre navigateur. Il faut penser à avoir la console des outils de développement systématiquement ouverte pour détecter des erreurs de code.



Première partie

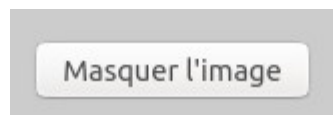
A) On souhaite inverser les contrastes lorsque l'utilisateur clique sur le bouton. Quelle sera la cible ? Quel sera l'événement ? Quelle sera l'action ?

B) Ecrire les instructions JS pour ajouter la classe « dark » à l'élément <body>, lorsque l'utilisateur clique sur le bouton.

C) Modifier les instructions JS pour retirer la classe « dark » à l'élément <body> si il a déjà cette classe au moment où se produit le clic.

Seconde partie

D) Dans le code html, ajouter un bouton pour masquer l'image. On souhaite masquer l'image lorsque l'utilisateur clique sur le bouton. Quelle sera la cible ? Quel sera l'événement ? Quelle sera l'action ?



E) Ecrire les instructions JS pour ajouter le style display « none » à l'image, lorsque l'utilisateur clique sur ce bouton.

F) Ecrire les instructions JS pour ajouter le style display «block» à l'image, lorsque l'utilisateur clique sur ce bouton et que l'image est invisible.

G) Modifier le texte du bouton en fonction de la situation.

Troisième partie

H) Ajouter un curseur html sous les deux boutons et constater qu'il s'affiche bien.

```
<input type="range" value="1" min="0" max="100">
```

On souhaite utiliser ce curseur pour faire varier l'opacité de l'image. Quelle sera la cible ? Quel sera l'événement ? Quelle sera l'action ?

I) Ecrire les instructions JS pour faire varier l'opacité de l'image, lorsque l'utilisateur fait varier le curseur.