

Javascript

TD 2 – Gestion d'événements « click »

Fonctions

Exercice 1 : Fonction simple

Voici une fonction qui converti des degrés Celsius en degrés Farenheit:

```
function conversion(valeur){  
    return valeur * 1.8 + 32 ;  
}
```

Écrire un programme Javascript qui demande à l'utilisateur de saisir une température (en Celsius) et qui affiche **en console** la conversion en Farenheit.

Conseil : utiliser la fonction **prompt()** pour demander une saisie à l'utilisateur.

Exercice 2 : appeler une fonction dans un programme

A. Ecrire une fonction isValid() qui prend en paramètre une chaîne de caractère, qui retourne true si cette chaîne contient un « @ » et qui retourne false dans le cas contraire.

B. Ecrire un programme qui demande à l'utilisateur de saisir une adresse email (en utilisant la fonction prompt), qui appelle la fonction isValid() et affiche en console la validité de l'adresse.

Méthodologie

Exercice 3 : Méthodologie

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

```
const target = document.querySelector('h1');
```

```
target.addEventListener('click', action);
function action() {
  console.log( 'click sur le titre' );
}
```

l'événement	
la cible/target	
l'action associée	

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

l'événement	
la cible/target	
l'action associée	

```
const target = document.getElementById('mask');
target.addEventListener('click', () => {
  const img = document.getElementById('img2');
  img.style.display='none' ;
});
```

l'événement	
la cible/target	
l'action associée	

Exercice 4 : Syntaxe

Corriger les fautes dans les instructions suivantes

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

Programmation simple

Exercice 5 : Les soldes

On dispose du code html suivant :

```
<div id="soldes">
  <p>Soldes du 15 juin au 15 juillet</p>
  <a href="#">Ok, j'ai compris</a>
</div>
```

et du code css :

```
#soldes.hidden{
  display: none;
}
```

Mettre en place un gestionnaire d'événement de telle sorte que le div soit masqué lorsque l'utilisateur clique sur le lien.

Exercice 6 : Mot de passe

Transformer un input de type password en type texte à l'aide d'un bouton et rendre le mot de passe visible.

Exercice 7 : Thème jour/nuit

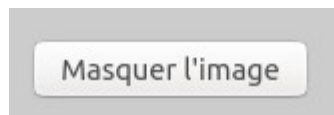
Créer un fichier html avec un titre, un bouton (« mode nuit ») et un bouton (« mode jour »). Associer un fichier css. Centrer les éléments sur la page. On souhaite inverser les contrastes lorsque l'utilisateur clique sur les boutons.

- Quelle seront les cibles ? Quels seront les événements ? Quels seront les actions ?
- Ecrire les instructions JS pour ajouter la classe « dark » à l'élément <body>, lorsque l'utilisateur clique sur le bouton « nuit ». Compléter le css, pour que le style de la page soit modifié : écriture claire et fond foncé.

- Modifier les instructions JS pour retirer la classe « dark » à l'élément <body> lorsque l'utilisateur clique sur le bouton « jour ».

Exercice 8 : Masquer l'image

Créer un fichier html avec un titre, une image et un bouton. Associer un fichier css. Centrer les éléments sur la page.



On souhaite masquer l'image lorsque l'utilisateur clique sur le bouton. Quelle sera la cible ? Quel sera l'événement ? Quelle sera l'action ?

- Ecrire les instructions JS pour ajouter le style display « none » à l'image, lorsque l'utilisateur clique sur ce bouton.
- 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.
- Modifier le texte du bouton en fonction de la situation (Afficher l'image / Masque l'image)

Plusieurs écouteurs

Exercice 9 : Formulaire de login

Connexion à votre compte particulier

Identifiant (10 chiffres)

Mémoriser mon identifiant

☐

Mot de passe (6 chiffres)

○ ○ ○ ○ ○ ○

4	7	9	3	0
6	8	2	1	5

[Identifiant / Mot de passe oublié](#)

En reprenant le code html fait précédemment, écrire le code JS pour le comportement suivant :

- au clic sur chaque bouton de chiffre, griser une puce.

Pour aller plus loin

- proposer un bouton pour réinitialiser le mot de passe (celui-ci s'affiche uniquement lorsqu'au moins un chiffre est cliqué)
- désactiver par défaut le bouton « se connecter ». L'activer uniquement lorsque les 6 puces sont grisées.

Exercice 10 : FAQ

Récupérer le fichier `faq.html`. Créer un script qui permet d'afficher une réponse lorsque l'utilisateur clique sur une question.

Temporisation

Exercice 11 : Temporisation d'événements

Créer un bouton et placer un écouteur d'événement : au clic, attendre 3 secondes et afficher un message en console. Si l'utilisateur clique plusieurs fois sur le bouton avant l'affichage du message, adapter le code de façon, à ce que seul le dernier clic soit pris en compte. **Conseil** : s'appuyer sur la fonction `clearTimeout()` : <https://developer.mozilla.org/en-US/docs/Web/API/Window/clearTimeout>