

TP 3

Gestion d'événements

Récupérer le dossier zippé TP3.zip ; le décompresser et ouvrir les fichiers dans un éditeur de code :
- index.html
- app.css
- app.js

Pour réaliser les exercices suivants, le code JavaScript est à placer dans le fichier app.js. Les autres fichiers ne doivent pas être modifiés.

Exercice 1

Menu « responsive »

Difficulté : 1

Objectif

Afficher / Masquer le menu, pour les formats d'écrans inférieurs à 600px de large.

Etape 1

Afficher la page en format 600px. Une icône « hamburger » s'affiche à la place du menu.

Etape 2

Dessiner le DOM situé sous l'élément <nav>

Etape 3

Observer le code CSS. Constaté qu'il suffit d'ajouter la classe 'responsive' à l'élément <nav> pour qu'il ait un affichage adapté aux mobiles.

Etape 4

On veut afficher le menu lorsque l'utilisateur clique sur l'icone « hamburger ». Quelle est la cible, l'événement et l'action.

Etape 5

Mettre en place un gestionnaire d'événement sur l'icone, de façon à afficher un message dans la console « élément cliqué »

Etape 6

Modifier le code, de façon à ajouter la classe « responsive », sur l'élément nav.

Etape 7

Pour fermer le menu, on retire la classe « responsive » à l'élément <nav> lorsque l'utilisateur clique sur l'icone #closeIcon. Mettre en place ce gestionnaire d'événement

Exercice 1

Menu déroulant

Difficulté : 4

Objectif

Lorsque l'utilisateur clique sur un élément du menu : afficher le sous-menu.

Etape 1

Dessiner le DOM situé sous l'élément <nav>

Etape 2

Observer le code CSS. Constaté qu'il suffit d'ajouter la classe 'visible' au ul.level-2 pour le faire apparaître.

```
#header nav ul.level-2.visible{
    display:block
}
```

Etape 3

Déterminer quelles sont les cibles, l'événement et les actions.

Etape 4

Mettre en place un gestionnaire d'événement sur chaque élément de menu, de façon à afficher un message dans la console « élément cliqué »

Conseil : penser à utiliser une boucle « for ».

Etape 5

Modifier la fonction pour récupérer dans une variable « **rubrique** », l'élément cliqué. Utiliser la propriété **target** expliquée dans le cours page 17.

Etape 6

Récupérer ensuite dans une variable « **menu** » le sous-menu à afficher.

Etape 7

Ajouter la classe « visible » à l'élément html représenté par la variable « **menu** ».

Etape 8

Modifier le code pour replier le menu, lorsque l'utilisateur clique une seconde fois

Etape 9

Améliorer le code en masquant systématiquement le sous-menu ouvert, avant d'en ouvrir un nouveau

Exercice 3

Scroll

Difficulté : 2

Objectif :

Au clic sur la flèche a#scrollDown, faire glisser la fenêtre du navigateur vers le bas. Et au clic sur la flèche a#scrollUp, faire glisser la fenêtre vers le haut.

Etape 1

Déterminer quelle est la cible, l'événement et l'action. Ecrire un gestionnaire d'événement qui affiche un simple message an console au clic sur la première flèche.

Etape 2

Pour faire glisser la fenêtre, on utilise la méthode window.scroll(x,y).

Consulter la documentation : <https://developer.mozilla.org/en-US/docs/Web/API/Window/scroll>

Créer un effet de glissement de 200 pixels vers le bas pour tester.

Etape 3

On veut un effet de glissement sur 100 % de la hauteur de la fenêtre du navigateur.

Celle-ci est accessible grâce à la propriété window.innerHeight :

<https://developer.mozilla.org/fr/docs/Web/API/Window/innerHeight>

Modifier le code en conséquence.

Etape 4

Implémenter le gestionnaire d'événement sur le second bouton.

Exercice 4

Afficher toute la liste / Réduire la liste

Difficulté : 3

Objectif

Si l'on observe le code html, on peut voir que le `div#categories` contient deux `div`. Le premier `div` (celui qui a la classe `liste-1`) est visible, le second (qui a la classe `liste-2`) est invisible. Le but est d'afficher le second `div`, lorsque l'utilisateur clic sur le bouton `a#toggle`.

Etape 1

Dessiner le DOM situé sous l'élément `div#categories`

Etape 2

Observer le code css, on peut voir qu'il suffit d'ajouter la classe 'visible' au `div.liste-2` pour le faire apparaître.

Etape 3

Déterminer quelle est la cible, l'événement et l'action. Ecrire un gestionnaire d'événement qui affiche un simple message an console au clic sur le bouton.

Etape 4

Dans le gestionnaire d'événement : ajouter la classe 'visible' au `div.liste-2` pour le faire apparaître.

Etape 3

Modifier le gestionnaire d'événement de façon à inverser le comportement : lorsqu'on clique une seconde fois sur le bouton, retirer la classe 'visible'.

Etape 5

Modifier le texte du bouton en conséquence : « Afficher toute la liste » / « Réduire la liste »