# M413 - TD1 : Réponses aux Questions

## Exercice 1

➢ Quel sera l’évènement qui déclenchera l’appelle de votre fonction ?

Le body appellera le code à faire charger grâce au "onLoad()"

➢ Quelle méthode avez-vous utilisée pour récupérer l’objet représentant votre balise "h1"?

J'ai utilisé la méthode getElementById()

*function* defineHeading1() {  
 *let* heading1 = document.getElementById('title');  
 heading1.innerHTML = 'TD1';  
}

➢ Quelle propriété de l’objet représentant votre balise « h1» avez-vous utilisée pour récupérer le texte de celui-ci ?

L'id

➢ Quelle(s) méthode(s) avez-vous utilisée pour récupérer l’objet représentant la première balise "h2" ?

J'ai utilisé la méthode getElementById()

*function* defineHeading2(){  
 *let* heading2 = document.getElementById('title2');  
 heading2.innerHTML = 'Exo 2';  
}

➢ Comment faire pour connaitre le nombre de balise « h2 » du document ?

J’ai utilisé un bouton afin de prendre tout les h2 et de regarder la taille du conteneur puis de la renvoyer

*function* countHeading() {  
 *var* h2 = document.getElementsByTagName('h2');  
 *var* h2Length = h2.length;  
 alert('Il y a ' + h2Length + ' balises h2 dans le document.');  
 }

➢ Quelle méthode avez-vous utilisée pour récupérer les objets de votre classe ?

J’ai utilisé la méthode getElementsByClassName()

➢ Comment avez-vous déterminé si un nombre est pair ?

J’ai regarder si le modulo des heading était pair ( heading4.length % 2 === 0 )

*function* defineHeading4() {  
 *let* heading4 = document.getElementsByClassName('firstOrLast');  
 *if* (heading4.length % 2 === 0) {  
 heading4[0].innerHTML = 'First Or Last ?';  
 } *else* {  
 heading4[heading4.length - 1].innerHTML = 'Exo 4';  
 }  
}

## Exercice 6.2

Quelles différences existe-t-il entre les 3 propriétés innerHTML, innerText et textContent ?

Les propriétés innerHTML, innerText et textContent sont toutes des propriétés des éléments HTML et permettent de lire ou de définir le contenu d'un élément. Cependant, elles ont des comportements différents :

innerHTML permet de lire ou de définir le contenu HTML d'un élément, y compris les balises HTML. C'est utile pour ajouter ou manipuler des éléments HTML enfants d'un élément.

innerText permet de lire ou de définir le contenu textuel d'un élément, mais ignore les balises HTML. C'est utile pour obtenir le contenu textuel d'un élément sans les balises HTML.

textContent permet de lire ou de définir le contenu textuel brut d'un élément, y compris les balises HTML. C'est utile pour obtenir le contenu textuel brut d'un élément, y compris les balises HTML.

En gros, si vous voulez manipuler le HTML d'un élément, utilisez innerHTML. Si vous voulez obtenir le contenu textuel brut d'un élément, utilisez textContent. Si vous voulez obtenir le contenu textuel d'un élément sans les balises HTML, utilisez innerText.

## Exercice 6.3

➢ Comment modifier votre code pour qu’il permette de sélectionner le 1 er auteur de la liste ?

<meta *name*="author" *content*="Auteur 1">  
<meta *name*="author" *content*="Auteur 2">  
<meta *name*="author" *content*="Auteur 3">  
<meta *name*="description" *content*="Description de la page">  
<meta *name*="keywords" *content*="mots clés">  
  
<div *id*="update-date"></div>  
  
<script>  
 *function* dateAlter() {  
 *var* lastModified = *new* Date(document.lastModified);  
 *var* date = lastModified.toLocaleDateString();  
 *var* author = document.querySelector("meta[name='author']").getAttribute("content");  
 document.querySelector("#update-date").innerHTML = "Dernière modification : " + date + " par " + author;  
 }  
  
 window.onload = dateAlter;  
</script>

➢ Même question avec le dernier auteur de la liste.

<meta *name*="author" *content*="Auteur 1">  
<meta *name*="author" *content*="Auteur 2">  
<meta *name*="author" *content*="Auteur 3">  
<meta *name*="description" *content*="Description de la page">  
<meta *name*="keywords" *content*="mots clés">  
  
<div *id*="update-date"></div>  
  
<script>  
 *function* dateAlter() {  
 *var* lastModified = *new* Date(document.lastModified);  
 *var* date = lastModified.toLocaleDateString();  
 *var* authors = document.querySelectorAll("meta[name='author']");  
 *var* author = authors[authors.length - 1].getAttribute("content");  
 document.querySelector("#update-date").innerHTML = "Dernière modification : " + date + " par " + author;  
 }  
  
 window.onload = dateAlter;  
</script>

## Exercice 7

➢ Comment obtenez-vous le nombre de jours ?

Pour obtenir le nombre de jours restants, on calcule la différence entre la date actuelle et la date d'échéance en utilisant la méthode getTime() de l'objet Date. Cette méthode retourne le nombre de millisecondes depuis le 1er janvier 1970 jusqu'à la date spécifiée. En divisant le résultat par la constante 86400000 (le nombre de millisecondes dans un jour), nous pouvons obtenir le nombre de jours entiers restants.

➢ Comment faites-vous la mise à jour du texte ?

Pour mettre à jour le texte, on utilise la propriété innerHTML de l'élément HTML cible. Cela nous permet de changer le contenu du paragraphe en ajoutant du texte à la suite du contenu existant, ou en le remplaçant complètement si nécessaire.

## Exercice 7.1

➢ Laquelle des deux méthodes de l’objet window avez-vous utilisé ? Pourquoi ?

J'ai utilisé la méthode setInterval de l'objet window. Cette méthode est utilisée pour exécuter une fonction ou une instruction de manière répétée à des intervalles réguliers.

Je l'ai utilisée pour mettre à jour l'heure affichée dans la balise <p> toutes les secondes, afin de toujours afficher l'heure actuelle. La méthode setInterval prend en paramètre la fonction à exécuter et l'intervalle en millisecondes entre chaque exécution.

## Exercice 8.1

➢ Quel évènement avez-vous utilisé ?

J'ai utilisé l'événement input pour détecter lorsque l'utilisateur saisit du texte dans le champ. Ensuite, j'ai utilisé JavaScript pour vérifier si la valeur entrée est un nombre ou non en utilisant la méthode isNaN() de JavaScript. Enfin, j'ai utilisé JavaScript pour ajouter ou enlever des classes CSS sur le champ de saisie en fonction de si la valeur entrée est un nombre ou non.

➢ Comment avez-vous fait changer la couleur du champ texte ?

Pour changer la couleur du champ texte, j'ai utilisé des instructions CSS pour définir les couleurs de fond pour les différentes classes CSS. Ensuite, j'ai utilisé JavaScript pour ajouter ou enlever des classes CSS sur le champ de saisie en fonction de si la valeur entrée est un nombre ou non.

## Exercice 8.2