

TP1 : JS Programmation web – client

http://users.polytech.unice.fr/~rey/cours/supports/js/TP1_JS_M413_2016.pdf

HETYEI Chloé

EXERCICE 1 :

1.1.

- L'évènement qui déclenchera l'appel de ma fonction sera l'évènement « onLoad ».
- Pour récupérer l'objet représentant ma balise h1, j'ai utilisé la méthode `document.getElementById('titre')`.
(Car nous avons défini un id sur notre h1 s'appelant titre).
- Pour récupérer le texte de ma balise h1, j'ai utilisé la propriété `innerText`.
- Pour récupérer l'objet représentant ma balise h2 j'ai utilisé la méthode `document.querySelector('h2')` puis `innerText` pour récupérer le texte.
- Pour connaître le nombre de balise h2 du document on peut faire `document.querySelectorAll('h2').length`
- Pour récupérer l'objet de ma classe j'ai utilisé la méthode `document.getElementsByClassName('firstOrLast')`
Car nous avons défini une classe `firstOrLast` dans la balise h1, la deuxième h2 et la première h3.
- Pour déterminer si un nombre est pair j'ai fait `(nombre % 2) == 0`.

1.2.

- Les différences entre ces 5 propriétés sont que
 - `innerHTML` est le plus utile et est supporté par tous les navigateurs. Il récupère le contenu entre deux balises avec d'éventuelles balises contenues dans le texte.
 - `InnerText` récupère le contenu entre deux balises sans interprétation d'éventuelles balises contenues dans le texte.
 - `OuterHTML` récupère tout le contenu entre deux balises (aussi les éventuelles balises contenues dans le texte) en récupérant celles-ci également.
 - `OuterText` est identique à `innerText` sauf que cette propriété n'est pas standard et n'est pas supporté par Firefox.
 - `TextContent` est dans la spécification standard et supporté par tous les navigateurs, cependant il lit uniquement du texte et n'est pas analysé en HTML comme peut le faire `innerHTML`.
- Il y a bien une différence car ces propriétés ne sont pas toujours supportées par tous les navigateurs. (Comme `innerText` et `outerText` qui ne sont pas supportées par Firefox)

1.3.

- Pour pouvoir sélectionner le premier auteur de la liste on peut faire `document.querySelectorAll('meta[name=author]')` qui renvoie la liste puis sélectionner le premier à l'index 0.
- Même réponse que précédemment mais en sélectionnant `index.length - 1`.

EXERCICE 2 :

1.0.

- On obtient le nombre de jours en faisant la différence en millisecondes entre la date du jour (`new Date()`) et la date indiquée, et on divise le résultat par la durée en millisecondes d'un jour.
- On fait la mise à jour du texte en ajoutant un événement `onClick` qui renvoie à la fonction `majNbJours` sur le texte qu'il faut mettre à jour.

1.1.

- Je préfère utiliser la méthode `setInterval`, car elle exécute la fonction affichant l'heure en bouclant tous les x temps pour mettre à jour l'heure. Tandis que l'autre méthode exécute la fonction qu'une seule fois.

EXERCICE 3 :

1.1.

- J'ai utilisé l'évènement `oninput`
- J'ai fait changer la couleur du champ texte en changeant le nom de la classe grâce à l'élément « `classname` » qui renomme la classe.