

PROGRAMMATION WEB AVANCEE

TP2

D.O.M

Exercice 1 :

Utiliser une structure DOM pour écrire une page web permettant de créer un lien pointant vers la page de www.gmail.com en cliquant sur le texte Lien vers gmail, et ce de deux manières :

- a) en utilisant `getElementsByName()`
- b) en utilisant `querySelector()`

Exercice 2 :

1. Créer une page html5 faisant apparaître une image quelconque.
2. Ajouter un bouton et le code JavaScript associé qui la fait disparaître (utiliser la propriété CSS de l'image `visibility`).
3. Ajouter un bouton qui fait revenir l'image.
4. Programmer un effet de fondu pour que l'image apparaisse et disparaisse progressivement (utiliser cette fois la propriété CSS de l'image `opacity`)

Exercice 3 :

L'objectif de cet exercice consiste à pouvoir accéder aux éléments à partir de l'ensemble du document. On tentera de travailler avec les méthodes `getElementById`, `getElementsByName` et `getElementsByTagName`.

Travail demandé :

1. Créer une fonction **modif_paragraphe**, appelée en cliquant sur le titre. Cette fonction sélectionne le paragraphe en utilisant son identifiant, puis le modifie avec la propriété **textContent** , en remplaçant le mot **original** en caractères droit par le mot **corrigé**, en caractère italique.
2. Créer une fonction **centrage_h1**, appelée en cliquant sur le premier paragraphe. Cette fonction détecte d'abord les éléments portant le nom de balise **h1**. Elle sélectionne ensuite le premier d'entre eux, dans notre cas, il n'y en a qu'un seul paragraphe sur la page, et modifie son attribut **align**, en lui affectant la valeur "center", à l'aide de la méthode **setAttribute**.

Exercice 4 :

Soit le fichier :

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Mon titre</title>
<script type="text/javascript" src="lib.js"></script>
</head>
<body>
<h1 onClick="showHxContent(1)">Partie 1</h1>
<div id="title1" style="display:none">
<p>Mon premier paragraphe</p>
<p>Mon second paragraphe</p>
</div>
<h1 onClick="showHxContent(2)">Partie 2</h1>
<div id="title2" style="display:none">
<p>Mon premier paragraphe</p>
<p>Mon second paragraphe</p>
</div>
<p>
<input type="submit" value="Masquer" onClick="hideAllDivs()"/>
</p>
<p>
Titre n°: <input type="text" id="title"/>
<input type="submit" value="Afficher" onClick="alertTitle()"/>
</p>
</body>
</html>
```

1. Écrire la fonction JavaScript showHxContent permettant d'afficher le contenu du div correspondant au titre sur lequel on clique.
2. Écrire la fonction JavaScript hideAllDivs permettant de masquer le contenu de tous les div du document.
3. Écrire la fonction alertTitle permettant d'afficher (avec la fonction JavaScript alert) le contenu du énième titre, défini par le champ de saisie title.
4. Écrire la fonction deleteTitle permettant d'effacer le contenu du titre défini par le champ title. (suppression du nœud fils de type texte)
5. Écrire la fonction defineTitle permettant d'affecter le contenu du titre défini par le champ title avec la valeur "Nouveau titre". On testera que le titre a ou non déjà une valeur (nœud texte) avant l'ajout, afin de la supprimer si besoin.

Exercice 5 :

L'objectif de cet exercice consiste à pouvoir changer les puces d'une liste non numérotée en

« circle » ainsi que les éléments de la liste par « Bonjour ». Pour cela, on vous demande de :

6. Récupérer l'objet de liste à l'aide d'une méthode `getElementsByTagName`.
7. Récupérer la collection d'items de liste à l'aide d'une méthode `getElementsByTagName`.
8. Changer la valeur de l'attribut de type de la liste en "circle".
9. Pour chaque élément de liste, changer le texte en "bonjour" et affecter à l'attribut `onclick` un appel à une boîte d'alerte avec le texte « Bon Travail »

Exercice 6 : gestion des évènements en javascript

Recopiez le code suivant afin de comprendre la propagation des évènements dans une page html :

```
<html onclick="alert('Propagation dans l\'élément html')">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>Event Bubbles</title>
<script type="text/javascript">
function init()
{
window.onclick = winEvent
document.onclick = docEvent;
document.body.onclick = docBodEvent;
}
function winEvent()
{
alert("Propagation dans l'objet window");
}
function docEvent()
{
alert("Propagation dans l'objet document");
}
function docBodEvent()
{
alert("La propagation se fait dans le BODY.");
}
</script>
</head>
<body onload="init()">
<h1>Propagation des évènements</h1>
<hr />
<form onclick="alert('Il se propage au formulaire.');">
<input type="button" value="Button 'main1'" name="main1" onclick="alert('L
évènement commence au bouton: ' + this.name)" />
</form>
</body>
</html>
```

Réécrire ce code en utilisant la méthode `addEventListener`

Pour stopper la propagation il faut ajouter la méthode `removeEventListener`

Essayer cette instruction à n'importe quel niveau dans l'exemple précédent.

Exercice 7 :

Un noeud du DOM, usuellement un élément HTML, peut s'inscrire pour traiter un événement à l'aide de sa méthode `addEventListener()` :

`ref_noeud.addEventListener(typeEvent, fctEvent, useCapture);`

où :

- **`ref_noeud`** est une variable JS qui désigne le noeud de type `Element` qui se porte candidat pour traiter l'événement.
- **`typeEvent`** : nom de l'événement (le même qu'en DOM 0 sans "on", click au lieu de `onclick`)
- **`fctEvent`** : nom de la fonction de traitement qui sera déclenchée à l'interception de l'événement
- **`useCapture`** : veut-on capturer l'événement ou seulement le déceler, l'intercepter ?

Ecrire une page html dont la structure du document sera un `div` identifié par l'id `conteneur`. Cet élément possèdera un ensemble d'éléments de type `div` (représentant des boîtes)

Ecrire une fonction `init()` en javascript qui sera lancée à la fin du chargement de la page afin d'ajouter l'événement `click` sur chacune des boîtes. La fonction lancée aux différents clicks devra faire en sorte d'afficher le numéro de la boîte dans la méthode `alert()` ;